National Agricultural Higher Education Project (NAHEP)
Centre for Advanced Agricultural Science and Technology (CAAST) on Genomics Assisted Crop Improvement and Management

High Dimensional Genome Data Analysis by R and Open Source Tools

A Training Manual

Compiled by

A. R. Rao
Sanjeev Kumar
Soumen Pal
Prabina Kumar Meher
Tanmaya Kumar Sahu

Centre for Agricultural Bioinformatics (CABin)
ICAR-Indian Agricultural Statistics Research Institute
ICAR-Indian Agricultural Research Institute
New Delhi-110012

2019

Citation:
Rao A.R. et al. (2019) High Dimensional Genome Data Analysis by R and Open Source Tools – A Training Manual. ICAR-Indian Agricultural Statics Research Institute, New Delhi

World Bank – ICAR Funded National Agricultural Higher Education Project (NAHEP)
Centre for Advanced Agricultural Science and Technology (CAAST) on Genomics Assisted Crop Improvement and Management
Training Programme on "High Dimensional Genome Data Analysis by R and Open Source Tools" from November 01 -11 November, 2019
Center for Agricultural Bioinformatics(CABin), Indian Agricultural Statistics Research Institute, ICAR-IARI, New Delhi-110012

ISBN:

Course Director
Dr. A.R. RAO
Principal Scientist & Professor (Bioinformatics),
Centre for Agricultural Bioinformatics (CABin),
ICAR-Indian Agricultural Statistics Research Institute,
Library Avenue, Pusa, New Delhi-110012

Course Coordinators
Mr. SANJEEV KUMAR
Scientist, Centre for Agricultural Bioinformatics (CABin)
ICAR-IASRI, New Delhi-110012

Dr. SOUMEN PAL
Scientist, Division of Computer Applications, ICAR-IASRI,
New Delhi-110012

Dr. PRABINA KUMAR MEHER
Scientist, Division of Statistical Genetics, ICAR-IASRI,
New Delhi-110012

Course Associate
Dr. TANMAYA KUMAR SAHU
Research Associate, NAHEP-CAAST
Centre for Agricultural Bioinformatics (CABin)
ICAR-IASRI, New Delhi-110012

Published by:
NAHEP-Centre for Advanced Agricultural Science and Technology
ICAR-Indian Agricultural Statistics Research Institute, New Delhi
ICAR-Indian Agricultural Research Institute, New Delhi
http://nahep-caast.iari.res.in

World Bank – ICAR Funded
National Agricultural Higher Education Project (NAHEP)
Centre for Advanced Agricultural Science and Technology (CAAST)
on Genomics Assisted Crop Improvement and Management

About NAHEP-CAAST at ICAR-IARI, New Delhi

Centre for Advanced Agricultural Science and Technology (CAAST) is a new initiative and student centric subcomponent of World Bank sponsored **National Agricultural Higher Education Project (NAHEP)** granted to the Indian Council of Agricultural Research, New Delhi to provide a platform for strengthening educational and research activities of post graduate and doctoral students. The ICAR-Indian Agricultural Research Institute, New Delhi was selected by the NAHEP-CAAST programme. NAHEP sanctioned Rs 19.99 crores for the project on "**Genomic assisted crop improvement and management**" under CAAST programme. The project at IARI specifically aims at inculcating genomics education and skills among the students and enhancing the expertise of the faculty of IARI in the area of genomics.

Objectives

1. **To develop online teaching facility and online courses for enhancing the teaching and learning efficiency, and scientific communications skills**
2. **To develop and/or strengthen state-of-the art next-generation genomics and phenomics facilities for producing quality PG and Ph.D. students**
3. **To develop collaborative research programmes with institutes of international repute and industries in the area of genomics and phenomics**
4. **To enhance the skills of faculty and PG students of IARI and NARES**
5. **To generate and analyze big data in genomics and phenomics of crops, microbes and pests for genomics augmentation of crop improvement and management**

IARI's CAAST project is unique as it aimed at providing funding and training support to the M.Sc. and Ph.D. students from different disciplines who are working in the area of genomics. It will organize lectures and training programmes, and send IARI students and covering students from several disciplines. It will provide opportunities to the students and faculty to gain international exposure. Further, the project envisages developing a modern lab named as **Discovery Centre** that will serve as a common facility for students' research at ICAR-IARI.

Acknowledgements

1. Secretary, DARE and Director General, ICAR, New Delhi
2. Deputy Director General (Education), ICAR, New Delhi
3. Director, ICAR-IARI, New Delhi
4. Director, ICAR-IASRI, New Delhi
5. Joint Director (Research), ICAR-IARI, New Delhi
6. Dean & Joint Director (Education), ICAR- IARI, New Delhi
7. National Coordinators, NAHEP, ICAR, New Delhi
8. Head, Centre for Agricultural Bioinformatics (CABin), ICAR-IASRI, New Delhi
9. CAAST Team, ICAR-IARI, New Delhi
10. Staff & Students, ICAR-IASRI, New Delhi

World Bank – ICAR Funded
National Agricultural Higher Education Project (NAHEP)
Centre for Advanced Agricultural Science and Technology (CAAST)
on Genomics Assisted Crop Improvement and Management

Core-Team Members

Principal Investigator

Dr. Viswanathan Chinnusamy
Head and Principal Scientist
Division of Plant Physiology
ICAR-IARI, New Delhi 110012

Nodal Officer

Dr. K. M. Majaiah
Associate Dean
Post-Graduate School
ICAR-IARI, New Delhi 110012

Nodal Officer (Grievance Redressal)

Dr. K. Annapurna
Head and Principal Scientist
Division of Microbiology
ICAR-IARI, New Delhi 110012

S. No.	Name of the Faculty	Discipline	Institute
1.	Dr. Ashok K. Singh	Genetics	ICAR-IARI
2.	Dr. Vinod	Genetics	ICAR-IARI
3.	Dr. Gopala Krishnan S	Genetics	ICAR-IARI
4.	Dr. A. Kumar	Plant Pathology	ICAR-IARI
5.	Dr. T. K. Behera	Vegetable Science	ICAR-IARI
6.	Dr. R. N. Sahoo	Agricultural Physics	ICAR-IARI
7.	Dr. Alka Singh	Agricultural Economics	ICAR-IARI
8.	Dr. A. R. Rao	Bioinformatics	ICAR-IASRI
9.	Dr. R. C. Bhattacharya	Molecular Biology & Biotechnology	ICAR-NIPB
11.	Dr. R. Roy Burman	Agricultural Extension	ICAR-IARI

Associate Team

14.	Dr. Kumar Durgesh	Genetics	ICAR-IARI
15.	Dr. Ranjith K. Ellur	Genetics	ICAR-IARI
16.	Dr. N. Saini	Genetics	ICAR-IARI
17.	Dr. D. Vijay	Seed Science & Technology	ICAR-IARI
18.	Dr. Kishor Gaikwad	Molecular Biology & Biotechnology	ICAR-NIPB
19.	Dr. Mahesh Rao	Genetics	ICAR-NIPB
20.	Dr. Veena Gupta	Economic Botany	ICAR-NBPGR
21.	Dr. Era V. Malhotra	Molecular Biology & Biotechnology	ICAR-NBPGR
22.	Dr. Sudhir Kumar	Plant Physiology	ICAR-IARI
23.	Dr. R. Dhandapani	Plant Physiology	ICAR-IARI
24.	Dr. Lekshmy S. Nair	Plant Physiology	ICAR-IARI
25.	Dr. Madan Pal	Plant Physiology	ICAR-IARI
26.	Dr. Shelly Praveen	Biochemistry	ICAR-IARI
27.	Dr. Suresh Kumar	Biochemistry	ICAR-IARI
28.	Dr. Ranjeet R. Kumar	Biochemistry	ICAR-IARI
29.	Dr. S. K. Singh	Fruits & Horticultural Technology	ICAR-IARI
30.	Dr. Manish Srivastava	Fruits & Horticultural Technology	ICAR-IARI
31.	Dr. Amit Kumar Goswami	Fruits & Horticulture Technology	ICAR-IARI

Associate Team

S. No.	Name of the Faculty	Discipline	Institute
32.	Dr. Srawan Singh	Vegetable Science	ICAR-IARI
33.	Dr. Gograj S. Jat	Vegetable Science	ICAR-IARI
34.	D. Praveen Kumar Singh	Vegetable Science	ICAR-IARI
35.	Dr. V.K. Baranwal	Plant Pathology	ICAR-IARI
36.	Dr. (Ms.) Deeba Kamil	Plant Pathology	ICAR-IARI
37.	Dr. Vaibhav K. Singh	Plant Pathology	ICAR-IARI
38.	Dr. Uma Rao	Nematology	ICAR-IARI
39.	Dr. S. Subramanium	Entomology	ICAR-IARI
40.	Dr. M.K. Dhillon	Entomology	ICAR-IARI
41.	Dr. B. Ramakrishnan	Microbiology	ICAR-IARI
42.	Dr. V. Govindasamy	Microbiology	ICAR-IARI
43.	Dr. S.P. Datta	Soil Science & Agricultural Chemistry	ICAR-IARI
44.	Dr. R.N. Padaria	Agricultural Extension	ICAR-IARI
45.	Dr Satyapriya	Agricultural Extension	ICAR-IARI
46.	Dr. Sudeep Marwaha	Computer Application	ICAR-IASRI
47.	Dr. Seema Jaggi	Agricultural Statistics	ICAR-IASRI
48.	Dr. Anindita Datta	Agricultural Statistics	ICAR-IASRI
49.	Dr. Soumen Pal	Computer Application	ICAR-IASRI
50.	Dr. Sanjeev Kumar	Bioinformatics	ICAR-IASRI
51.	Dr. S.K. Jha	Food Science & Post Harvest Technology	ICAR-IARI
52.	Dr. Shiv Dhar Mishra	Agronomy	ICAR-IARI
53.	Dr. D.K. Singh	Agricultural Engineering	ICAR-IARI
54.	Dr. S. Naresh Kumar	Environmental Sciences	ICAR-IARI

Preface

Driven by the recent developments in plant/animal/fish genome projects, bioinformatics and genomics are taking an ever increasing role in agricultural sciences. In a similar way, advances in information technology and computational methods are driving the mathematical sciences forward. Now with the availability of new genome sequencing technologies, advanced statistical techniques and powerful computational algorithms, it is possible to handle and analyze very high dimensional omics data (genomic, proteomic and phenomic). Moreover, R and other open source tools are highly useful to analyze high-dimensional data emerging form genomics and phenomics projects. Thus an integrated approach is essential at these cross-roads to understand the underlying biological phenomena of complex traits. Above all, there exists always a rapid increasing demand for individuals to be trained in bioinformatics with special skills and knowledge to handle high dimensional genome data. Keeping this in view, the present training programme on "**High Dimensional Genome Data Analysis by R and Open Source Tools"** is being organized for the students of SAUs/CAUs/ICAR-DUs pursuing post graduate degree programmes in Agriculture across the country.

The statistical and computational algorithms and approaches discussed in this reference manual will acquaint the participants with the recent advances taken place in the field of *High Dimensional Genome Data Analysis*. This training will help the students in upgrading their analytical skills, especially in the area of ***Genomics Assisted Crop Improvement and Management***. Knowledge acquired during this training programme will be immensely helpful to the young participants to enhance their preparedness to tackle future challenges in the fields of genomics, proteomics and phenomics. Thus, the present training is being organized with the aim to (i) train post graduate students in handling high dimensional -omics data (ii) familiarize the students with R and other open source software for analysis of -omics data (iii) upgrade analytical skills of the participants.

Here, we wish to express our heartfelt gratitude to the faculty members, drawn from within and outside the Institute, for providing us with the reading materials in time. Our best wishes to all the students who have come from different SAUs/CAUs/ICAR-DUs, without whom conduct of the course would not have been a reality. Our sincere thanks and acknowledgements to the World Bank Funded ***National Agricultural Higher Education Project (NAHEP)*** for providing us the funds to organize the training programme under the sanctioned project: Genomics Assisted Crop Improvement and Management. The help support and guidance provided from time to time by our Research Managers from the Council; National Director and National Coordinator (CAAST) of NAHEP; Director, Joint Director (R), Dean & Joint Director (Education), Dr. Viswanathan, C., PI, Core Team Members and Associated Members of NAHEP-CAAST project implemented at IARI, Director, IASRI and Head, Centre for Agricultural Bioinformatics are sincerely acknowledged. We are particularly grateful to Mrs. Manjeet Kaur, Chief Technical Officer, Dr. Tanmaya Kumar Sahu and Mrs. Sarika Sahu, Research Associates, Mrs. Shivangi Varshney, Senior Research Fellow, Mr. Subhrajit Satpathy, Ph.D. scholar for helping us in conducting this training programme.

We gratefully acknowledge the infrastructure facilities – Computer labs, technical and administrative manpower, Guesthouse facilities, *etc.* provided by IARI and IASRI, New Delhi.

A.R.Rao
Sanjeev Kumar
Soumen Pal
P.K. Meher

31st October, 2019

NAHEP sponsored Training programme
on
"High Dimensional Genome Data Analysis by R and Open Source Tools"
(November 01-11, 2019)

Class Schedule

Day 1	**Friday, 01st November, 2019**	
9:30-10:00	Registration	
10:00-11:30	High Dimensional Genome Data Analysis –An overview	Dr. A.R. Rao
11:45-13:15	Overview on R and R-Studio (Basics commands & Data handling)	Dr. Soumen Pal
14:15-16:45	Practical on Descriptive Statistics and Linear models using R	Dr. Soumen Pal
17:00-18:30	Practical on Transcriptomics Data analysis	Dr. A.R. Rao and Mrs. Sarika Sahu
Day 2	**Saturday, 02nd November, 2019**	
09:30-11:30	**Inauguration Programme:**	
	Lightning of Lamp	
	Welcome Address: Director, ICAR-IASRI	
	About Training Programme: Dr A. R. Rao, Course Director, Core Team Member, NAHEP-CAAST, ICAR-IASRI	
	Students' and Faculty introduction	
	Remarks: Dr. A. K. Singh, Director, ICAR-IARI	
	Remarks: Dr. R. C. Agrawal, National Director, NAHEP & DDG Education, ICAR	
	Inaugural Address: Dr. T. Mohapatra, Secretary, DARE & DG, ICAR	
	Vote of thanks: Sh Sanjeev Kumar, Course Coordinator, Associate Team Member, NAHEP-CAAST, ICAR-IASRI	
	Photo Session	
11:45-13:15	R-Graphics	Dr. P. K. Meher & Mr. S. Satpathy
14:15-16:45	Testing of Hypothesis and analysis Experimental Design	Dr. Seema Jaggi
	Practical on Testing of Hypothesis and analysis Experimental Design using R	Dr. B.N. Mandal
17:00-18:30	Visit to National Agricultural Science Museum, National Agricultural Science Centre campus, ICAR, New Delhi	
Sunday, 3rd November, 2019-HOLIDAY		
Day 3	**Monday, 4th November, 2019**	
10:00-11:30	Big Data Analytics for Bioinformatics	Dr. Dinesh Gupta
11:45-13:15	R for High Dimensional Genome Data Analysis	Dr. S. Bhattacharjee
14:15-16:45	R for Biomolecular Sequence Data	Dr. P. K. Meher, Dr. A. R. Rao & Dr. T. K. Sahu
17:00-18:30	Analysis of Molecular Variance	Dr. A. R. Rao

Day 4	**Tuesday, 5th November, 2019**	
10:00-11:30	Genome Assembly	Sh. Sanjeev Kumar
11:45-13:15	Genome Annotation	Sh. Sanjeev Kumar
14:15-16:45	Practical on Genome Assembly and Annotation	Sh. Sanjeev Kumar & Mrs. Sarika Sahu
17:00-18:30	Practical on Biomolecular Sequence Encoding	Dr. T. K. Sahu
Day 5	**Wednesday, 6th November, 2019**	
10:00-11:30	Genome Editing	Dr. Suresh Kumar
11:45-13:15	Machine learning techniques (SVM, ANN, RF and Clustering Techniques)	Sh. Sanjeev Kumar
14:15-16:45	Practical on Machine Learning using R	Sh. Sanjeev Kumar & Dr. T. K. Sahu
17:00-18:30	Genome editing practical	Dr. A. R. Rao & Mrs. Sarika Sahu
Day 6	**Thursday, 7th November, 2019**	
10:00-11:30	Genome Wide Association Study (GWAS)-Statistical View Point	Dr. Anil Rai
11:45-13:15	Visit to Advanced Super-computing Hub for OMICS Knowledge in Agriculture (ASHOKA)	Dr. K. K. Chaturvedi
14:15-16:45	Principal Component Analysis and Discriminant Analysis	Dr. P. K. Meher
17:00-18:30	Practical on Phylogenetic Analysis	Dr. T. K. Sahu
Day 7	**Friday, 8th November, 2019**	
10:00-11:30	Genome Wide Association Study (GWAS)	Dr. S. Bhattacharjee
11:45-13:15	Practical on GWAS	Dr. S. Bhattacharjee
14:15-16:45	Practical on Protein Structure Prediction	Dr. T. K. Sahu
17:00-18:30	Practical on Genomic Selection and SNP Analysis	Dr. A. R. Rao, Mrs. Sarika Sahu & Mr. S. Satpathy
Day 7	**Saturday, 9th November, 2019**	
10:00-11:30	Application of Bioinformatics in Nematology	Dr. Uma Rao
11:45-13:15	Visit to Nanaji Deshmukh Plant Phenomics Centre, ICAR-IARI	
14:15-16:45	Genetic Analyses Of Quantitative Traits: Challenges And Caveats	Dr. Saurabh Ghosh
Sunday, 10th November, 2019-HOLIDAY		
Day 8	**Monday, 11th November, 2019**	
10:00-11:30	Plant Phenomics	Dr. Viswanathan C.
11:45-13:15	Practical on Phenomics Data Analysis	Sh Sanjeev Kumar
14:15-15:00	**Interaction and Post Training Evaluation**	
15:00-16:30	**Valedictory function and certificate distribution**	

11:30-11:45: Tea

13:15-14:15: Lunch

16:45-17:00: Tea

List of Resource Persons

S.No.	Name	Designation	Affiliation	email
External Resource Persons				
1.	Dr. Dinesh Gupta	Group leader	Translational Bioinformatics, International Centre for Genetic Engineering and Biotechnology, Aruna Asaf Ali Marg, New Delhi	dinesh@icgeb.res.in
2.	Dr. S. Bhattacharjee	Assistant Professor	National Institute of Biomedical Genomics, West-Bengal	sb1@nibmg.ac.in
3.	Dr Suresh	Principal Scientist	Division of Biochemistry, ICAR-Indian Agricultural Research Institute	sureshkumar@iari.res.in
4.	Dr. Uma Rao	Head & Principal Scientist	Division of Nematology, ICAR-Indian Agricultural Research Institute	umarao@iari.res.in
5.	Dr. Saurabh Ghosh	Professor	Human Genetics Unit, Biological Sciences Division, Indian Statistical Institute, Kolkata	saurabh@isical.ac.in
6.	Dr. Viswanathan C.	Head & Principal Scientist	Division of Plant Physiology, ICAR- Indian Agricultural Research Institute, New Delhi-12	viswanathan@iari.res.in
7.	Dr. Pritish Varadwaj	Associate Professor	Indian Institute of Information Technology Allahabad	pritish.varadwaj@gmail.com
Internal Resource Persons				
8.	Dr Anil Rai	Head	Centre for Agricultural Bioinformatics in ICAR-Indian Agricultural Statistics Research Institute, New Delhi	anil.rai@icar.gov.in
9.	Dr. Seema Jaggi	Head & Professor	Division of Design of Experiment, ICAR-Indian Agricultural Statistics Research Institute, New Delhi	seema.jaggi@icar.gov.in
10.	Dr. A.R. Rao	Principal Scientist & Professor	Centre for Agricultural Bioinformatics, ICAR-Indian Agricultural Statistics Research Institute, New Delhi	ar.rao@icar.gov.in
11.	Sh. Sanjeev Kumar	Scientist	Centre for Agricultural Bioinformatics, ICAR-Indian Agricultural Statistics Research Institute, New Delhi	sanjeev.kumar@icar.gov.in
12.	Dr. Soumen Pal	Scientist	Division of Computer Applications, ICAR-Indian Agricultural Statistics Research Institute, New Delhi	soumen.pal@icar.gov.in

13.	Dr. B.N. Mandal	Scientist	Division of Design of Experiments, ICAR-Indian Agricultural Statistics Research Institute, New Delhi	bn.mandal@icar.gov.in
14.	Dr. Prabina Kumar Meher	Scientist	Division of statistical Genetics, ICAR-Indian Agricultural Statistics Research Institute, New Delhi	prabina.meher@icar.gov.in
15.	Dr. Tanmaya Kumar Sahu	Research Associate	Centre for Agricultural Bioinformatics, ICAR-Indian Agricultural Statistics Research Institute, New Delhi	tanmayabioinfo@gmail.com
16.	Mrs. Sarika Sahu	Research Associate	Centre for Agricultural Bioinformatics, ICAR-Indian Agricultural Statistics Research Institute, New Delhi	sahusarikaiiita@gmail.com
17.	Mr. Subhrajit Satpathy	PhD. Scholar	Division of Agricultural Statistics, ICAR-Indian Agricultural Statistics Research Institute, New Delhi	satpathyiasri@gmail.com

Contents

High Dimensional Genome Data Analysis

Advances in information technology and computational methods have drived the mathematical sciences forward. In particular, Artificial Intelligence and Big Data analytics have revolutionized the bioinformatics-based genome data analysis. High-throughput genome technologies have made it possible to generate massive -omics data for understanding the complex phenomena involved in biological systems. With the availability of next generation sequencing (NGS) technologies, voluminous structured and unstructured data is now available. High-dimensional genomic data analysis is also challenging due to presence of noise and biases. Thus, computational analysis of such high-dimensional data often includes identification and correction of hidden biases, dimensionality reduction and application of AI / machine learning techniques for mining the hidden information to answer unsolved problems. Moreover, combined analysis of high dimensional genomic data from various sources to interpret the biological insights is highly difficult. High-dimensional genomic data usually presented as a matrix, with each column representing a sample and each row representing a genomic feature (for example, a gene, a genomic locus and so on). By computational analyses of these high-dimensional data matrices using dimension reduction techniques like, principal component analysis or clustering approaches, can help learn characteristic information within samples and identify key features between samples to interrogate biological functions. Computational methods like, Matrix Analysis and Normalization By Concordant Information Enhancement (MANCIE) that uses Bayesian-supported principal component analysis-based approach to adjust the data for bias correction and data integration of distinct genomic profiles on the same samples are now available in literature. Such methods can improve tissue-specific clustering, prognostic prediction in Molecular Taxonomy, copy number and expression agreement in Cell Line data, and has broad applications in high-dimensional data integration.

With the advent of cost effective technologies, more than one data matrices can be generated from multiple platforms of experiments on the same set of samples. Integrative analysis of such data sets is critical for obtaining biological insights, within which a common challenge exists in identifying and correcting hidden biases. Several methods have been developed to remove batch effect within one data matrix of the same platform. Sparse PCA uses the linear combination of a small subset of variables instead of all to generate the principal components and still explains most variances present in the data, while making the dimension reduction and bias removal clearer and easier to interpret. In a similar way, Surrogate variable analysis (SVA) models the gene-expression heterogeneity bias as ‘surrogate variables’ and separate them from primary variables that capture biologically meaningful information.

Statistical and computational analyses of biological sequences have completely changed their character since late 1980s. As more and more genome projects are coming up the emphasis has now been shifted from accumulation of such data to its interpretation. Statistical and computational tools for classifying sequences, detecting similarities, discriminating protein coding regions from non-coding regions in DNA sequences, identification of transcription binding sites, predicting molecular structure, molecular marker based classification of genotypes, etc, have become an essential component of biological research process. Bioinformatics is emerging at the frontier as an integrated discipline among biology, computer science and statistics, impacting medicine, agriculture, biotechnology, and society in many ways. Large amount of biological information has created both challenging data mining problems and opportunities, each requiring new ideas. Due to complex nature of biological system the conventional computer algorithms are unable to address many of the important sequence analysis problems. Artificial Intelligence, in general, and Machine Learning and deep learning, in specific, are ideally suited for domains characterized by the presence of large amounts of data, noisy patterns, and the absence of general theories. The fundamental idea behind these approaches is to learn the theory from data, through a process of inference, model fitting, or learning from examples. Thus they form a viable complementary approach to conventional methods.

Machine learning methods are computationally intensive and benefit greatly from progress in computer speed. To the novice, machine learning methods may appear as a bag of unrelated techniques. On the theoretical side, a unifying framework for all machine learning methods has also emerged since the late 1980s. This is the Bayesian probabilistic framework for modelling and inference. In fact, it is the confluence of all the three factors – data, computers and statistical theory – that is fuelling the machine learning expansion in high dimensional genome data analysis and elsewhere. An often met criticism of machine learning techniques is that they are "back box" approaches: one can not always pin down exactly how a complex Neural Network or Hidden Markov Model (HMM), reaches a particular answer. It is important to realize, however, that many other techniques in contemporary molecular biology are used on a purely empirical basis. Once a parameterized model M(w) for the data has been constructed, machine learning algorithms can be explained with the steps: (i) the estimation of the complete distribution P(w,D) and the posterior P(w|D) (ii) the estimation of the optimal set of parameters w by maximizing P(w|D), the first level of Bayesian inference (iii) The estimation of marginals and expectations with respect to the posterior, that is, for instance, of integrals of the form $E(f) = \int f(w)P(w \mid D)dw;$ the higher levels of Bayesian inference.

Machine learning is the analysis step in the process of Knowledge Discovery in Databases that results in the discovery of new patterns in large data sets. Various machine learning related techniques are Artificial Neural Network (ANN), Hidden Markov Model (HMM), Gibbs sampling, Support Vector Machines (SVM) and Random Forest etc. These techniques are

extensively used in genomics research. HMMs are very well suited for many tasks in molecular biology. HMMs are similar to Markov chain, but are more general and flexible, and allows to model phenomena that cannot be explained well with a regular Markov chain model. The most popular use of HMM in molecular biology is as a 'probabilistic pro-file' of a protein family, which is called a profile HMM. From a family of proteins (or DNA) a profile HMM can be made to search a database for other members of the family. HMMs are suitable for gene finding where several signals must be recognized and combined into a prediction of exons and introns, and the prediction must conform to various rules to make it a reasonable gene prediction.

Completion of human, plant and animal genomes have demonstrated that genomic sequence is the most comprehensive way towards gene discovery - a first step towards identifying the role of each gene. Comprehensive understanding of gene function will require thorough investigations of their own genomes to identify all of their genes and to determine the function of those genes. However, SNPs are distributed widely over these genes and the functions of these genes are associated with the population characteristics. As the next generation sequencing technologies are available, it becomes possible to get genome wide SNPs data and identify the SNPs associated with the disease response as well as genome based predictions of economically important traits. The major problem to be tackled in such genome wide SNPs data or Genotype-by-Sequencing (GBS) data is that the number of SNPs (p) is much larger than the number of subjects or indiciduals present in the experiment. In statistical theory, this problem is popularly referred as $p>>n$.

Genome Wide Association Study (GWAS)

GWAS is an examination of all or maximum of genes in different individuals of a particular species to variations among individuals. Different variations are then associated with different traits, such as diseases. If the genetic variations are more frequent in a population with a disease, the variations are said to be "associated" with the disease. The associated genetic variations are then considered as pointers to the region of the genome under study where the disease-causing problem is likely to reside. Two methods are used to search for disease-associated mutations: (i) hypothesis-driven methods, start with the hypothesis that a particular gene may be associated with a particular disease, and tries to find the association and (ii) non-hypothesis-driven (hypothesis generation) studies use brute force methods to scan the entire genome to identify those genes demonstrate an association. GWAS is generally non-hypothesis-driven.

In recent years, complex trait research has witnessed yet another revolution with the introduction of high density SNP arrays for enabling genome wide genotyping and whole genome association studies. Besides, Copy Number Variations (CNVs) in the genome associated with disease can also be identified from the high density SNP arrays. A systematic search and molecular characterization of CNVs are expected to provide useful insights into

their role in diseases. It is being increasingly recognized that careful phenotypic characterization together with statistical approaches are critical for the outcome of WGA studies. Thus, identification of disease causing genetic variants, understanding their interactions with other genes or pathways by statistical techniques helps in maintaining health care.

Genome prediction

The Mendelian rules of genetic inheritance are most obvious when the traits are controlled by a single gene. However, most of the economically and medically important traits in animals are complex in nature. As an example, most of the complex diseases are governed by a large number of loci with small and cumulative genetic effects. In such situations, predicting risk is a quite challenging task. Recently, Genome Prediction (GP) technique has been used to identify the genetic variants that can contribute to the risk prediction of the phenotypes.

Genome Prediction Methods

Let Y_i be a binary indicator for the phenotype of subject $i=1,\ldots,n$, and let X_{ij} be the estimated allele dosage of SNP $j=1,\ldots,p$ for subject i. Further, it is considered $Y=(Y_1,\ldots,Y_n)^t$, $X_j=(X_{1j},\ldots,X_{nj})^t$, and $x_i=(X_{i1},\ldots,X_{ip})^t$. Penalized regression technique is carried out on data set of the p most significant SNPs. These most significant predictors are pre-selected on just the training data, and this selection is repeated each of the cross-validation steps.

Since for most phenotypes that are studied in GWAS the signal is small, and often other risk factors are known, it is focused here mainly on modeling the probability that a subject is a case using logistic regression, rather than looking at classification. If a classification rule is needed probabilistic estimates can be thresholded taking mis-classification costs in consideration. The simplest approach to model the probabilities is to fit a linear logistic regression model on the p pre-selected SNPs:

$$\text{logit}(P(Y_i = 1|x_i)) = \beta_0 + \sum_{j=1}^{p} \beta_j X_{ij} + \varepsilon_i$$

Traditionally, parameters in this model are estimated using maximum likelihood. When large numbers of predictors are used, the logistic regression model is known to overfit the data. Instead, Least Absolute Shrinkage Selection Operator (LASSO) and *elasticnet*, two examples of penalized regression methods are considered for the study.

Let $l(\beta; Y_i, x_i, i=1,\ldots,n)$ be the logistic log-likelihood. The LASSO and elasticnet estimates of β are the maximizers of

$$l(\beta; Y_i, x_i, i=1,\ldots,n) - \lambda_1 \sum_{j=1}^{p} |\beta_j| \text{ and}$$

$$l(\beta; Y_i, x_i, i=1,\ldots,n) - \lambda_1 \sum_{j=1}^{p} |\beta_j| - \lambda_2 \sum_{j=1}^{p} \beta_j^2 ,$$

respectively, where λ_1 and λ_2 are selected using cross-validation. Both these approaches effectively carry out model selection, as the l_1 penalty $\lambda_1 \sum_{j=1}^{p} |\beta_j|$ will set many of the coefficients β_j to 0. The potential advantage of the elastic net is that when many of the predictors are highly correlated, the l_2 penalty $\lambda_2 \sum_{j=1}^{p} \beta_j^2$ encourages averaging of multiple-correlated predictors, while the lasso would select just a single predictor. The elastic net penalty can be viewed as a combination of the lasso penalty and the l_2 penalty form ridge regression, an early penalized regression method. Predicted probabilities of disease are evaluated using the logistic regression model on the test data. The fitted probabilities are summarized using the test data log-likelihood, receiver operating characteristic (ROC) curves and the Area Under Curve (AUC).

Cross-validation and the selection of significant predictors

Cross-validation is a technique for assessing how the results of a statistical analysis will generalize to an independent data set. It is mainly used in settings where the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice. One round of cross-validation involves partitioning a sample of data into complementary subsets, performing the analysis on one subset (called the training set), and validating the analysis on the other subset (called the validation set or testing set). To reduce variability, multiple rounds of cross-validation are performed using different partitions, and the validation results are averaged over the rounds. Hence, in K-fold cross-validation, the original sample is randomly partitioned into K subsamples. Of the K subsamples, a single subsample is retained as the validation data for testing the model, and the remaining K − 1 subsamples are used as training data. The cross-validation process is then repeated K times (the folds), with each of the K subsamples used exactly once as the validation data. The K results from the folds then can be averaged (or otherwise combined) to produce a single estimation. 10-fold cross-validation is the most commonly used cross-validation technique. In stratified K-fold cross-validation, the folds are selected so that the mean response value is approximately equal in all the folds. In the case of a dichotomous classification, this means that each fold contains roughly the same proportions of the two types of class labels. Using the packages glmnet and cv.glmnet of R software, LASSO model can be implemented on the data.

Feature generation and feature selection in different online tools

One of the major challenges in sequencing projects is to convert the textual (nucleotide or amino acid residue) information into numerical information so that numerical vectors can be used in machine learning approaches like ANN, SVM and Random Forest or in deep learning methods like convolutional neural networks (CNN). Such conversion leads to the generation of features (variables) and selection of variables is an importatnt step in the application of learning approaches. In case of GBS data, the genome wide SNPs are treated as features or variables. Selection of SNPs associated with trait under consideration is also a challenging task in high dimensional genome data analysis. The following are the few applications where fearture generations are outlined. With regards to the feature selection, the description on how to identify important features is given in the lecture notes on machine learning approaches.

HRGPred: Prediction of herbicide resistant genes with k-mer nucleotide compositional features

Here, two diferent types of k-mer features viz., contiguous k-mer (CkM) and pseudo k-mer (PkM) are considered. The CkM features have been used earlier for classifying the bacterial genomes, biological sequence clustering, predicting splicing junctions, DNA barcode-based species identification, *etc*. Also, the PkM features have been successfully used in many bioinformatics studies such as identifcation of DNA methylation sites, protein-protein interaction, N6-methyladenosine sites, RNA 5-methyl cytosine sites and prediction of protein sub mitochondrial locations.

funbarRF: DNA barcode-based fungal species prediction using multiclass Random Forest supervised learning model

The g-spaced base pair features are used to encode the barcode sequences into numeric feature vectors. Five kinds of g-spaced features namely 1-spaced (g=1), 2-spaced (g=2), 3-spaced (g=3), 4-spaced (g=4) and 5-spaced (g=5) were computed. For any nucleotide sequence of length N, each g-spaced feature-set results in 16 descriptors. The frequency of the di-nucleotide *s* and *t* with g-gap (g-spaced feature value) is given by $Dg(s, t)/(N - 1)$, where *s*, *t* = A, T, G, C; g = 1, 2, 3, 4, 5 and Dg(*s*, *t*) represents the counts of di-nucleotide *s* and *t* with g-gap. The g-spaced base pair features were computed by using *BioSeqClass R-package,* where the function feature *CKSAAP* was executed to generate the features.

nifPred: Proteome-Wide Identification and Categorization of Nitrogen-Fixation Proteins of Diaztrophs Based on Composition-Transition-Distribution Features Using Support Vector Machine

Six different sequence-based features to map the amino acid sequences into vectors of numeric observations were used. The features are compositions of amino acid residue compositions of di-peptides, pseudo amino acid compositions, composition-transition-distribution, gap-pair compositions, and auto-correlation function. Succinct descriptions about computation of the above mentioned features are given in the following sub-sections.

(a) Amino acid composition (AAC)

AAC is the simplest and most widely used feature for representing the protein sequences. It is nothing but the proportions of amino acid residues present in the sequence. Based on AAC, every protein sequence can be converted to a vector of 20 numeric observations. For a protein sequence with N residues, AAC for the ith amino acid can be computed as AAC(i) = fi/N, where i = 1, 2, …, 20 and fi indicates the number of times ith amino acid present in the sequence.

(b) Di-peptide composition (DPC)

Unlike AAC, DPC takes the ordering effects of amino acid residues within a short range into consideration (Ding et al., 2004). Anticipating improvement in accuracy by accounting the local-ordering of residues, DPC were considered as features. For any di-peptide Mj, DPC can be computed as DPC(j) = Mj/(N − 1), where j = 1, 2,…, 400 and N denotes the sequence length. Using DPC, each protein sequence can be transformed into a 400-dimensional numeric vector.

(c) Gap-pair composition (GPC)

For a given sequence with N amino acid residues, GPC for amino acid pair (i, j) with G-gap can be obtained as fG(i,j)=DG(i,j)(N−G−1), where i, j = 1, 2, …, 20 and DG(i, j) is the number of times the amino acid pair (i, j) appears in the sequence. Using GPC features, every amino acid sequence can be encapsulated with a numeric vector of 400 elements. Presently, we used 1 gap-pair (GPC-1) and 2 gap-pair (GPC-2) compositions as features. More clearly, for GPC-1 and GPC-2, the features are nothing but the proportions of amino acid pairs (i, j) separated by one residue (ixj), and two residues (ixxj) respectively, where x denotes any residue.

(d) Pseudo amino acid composition (PseAAC)

The PseAAC not only takes into account the sequence-ordering information within a local range but also the global sequence-ordering effects. This feature has been proven effective in many protein-related classifications (Wang et al., 2010). Using PseAAC, every protein sequence can be encoded to a (20+d)-dimension vector of numeric observations for d-tier correlation structure.

(e) Composition-transition-distribution (CTD)

In CTD, C (composition) stands for the compositions of amino acids, T (transition) represents the percentage with which frequency of amino acids with specific properties is followed by amino acids with other properties and D (distribution) determines the length of the sequence within which the 1st as well as 25, 50, and 75 percents of amino acids of certain characteristics are located. With CTD feature, each sequence of N amino acid residues can be encoded to a numeric vector of N+{N*N−12}+(N*5) elements.

(f) Auto-correlation function (ACF)

Auto-correlation takes into account the dependencies among sequence features, which are computed by taking the distribution of amino acid properties into account. Here, the ACF-based features were computed by considering all 531 amino acid properties obtained from AAindex database. Using ACF features, every sequence can be encoded to a (531*n)-dimensional vector of numeric observations, for nth order autocorrelation. Here, we considered the 1st order autocorrelation only, because with higher order number of features will be very large.

SPIDBAR: Identification of species by DNA barcode using k-mer feature vector and Random forest classifier

Here, frequencies of oligo-nucleotide strings of different scale k were used to map the barcode sequences onto numeric vectors.

For a given barcode sequence of length L, the number of possible oligo-nucleotide strings of R consecutive bases α1, α2, …, αR is (L – R + 1), where αi ϵ {A, T, G, C}. Let n(α1, α2, …, αR) be the number of times the string α1, α2, …, αR appears in the barcode sequence, by sliding through the sequence, shifting one nucleotide position at a time. Then, the probability of string α1, α2, …, αR appearing in the sequence can be computed as n(α1, α2, …, αR)/(L – R + 1). For instance, in the DNA sequence 'TGAGGTTTGTTTACGGTGAT', p(A) = 3/20, p(TT) = 4/(20 – 2 + 1) and p(TTT) = 2/(20 – 3 + 1). There are 4k and ∑k=a,b,c,...4k features possible for single scale k (i.e., k = 1 or 2 or 3, etc.) and multiple scale k (i.e., k = (1, 2), (1, 3), (1, 3, 4) etc.) respectively.

iAMPPred: Predicting antimicrobial peptides with improved accuracy

Since the peptide sequences are the strings of amino acids, they need to be mapped onto numeric feature vectors before being used as an input in supervised learning classifiers. Three different categories of features i.e., compositional, PHYC and STRL were considered here. In particular, 3 compositional (amino acid composition-AAC, pseudo amino acid composition-PAAC and normalized amino acid composition-NAAC), 3 PHYC (hydrophobicity, net-charge and iso-electric point) and 3 STRL (α-helix propensity, β-sheet

propensity and turn propensity) features were considered (Table 2) for prediction of AMPs. The compositional and PHYC features were computed by using the "***Peptide***" package of R-software, whereas the STRL features were computed by using the TANGO software available at http://tango.crg.es/. Furthermore, to know the importance of each feature in predicting the antibacterial, antiviral and antifungal peptides, information gain was computed for all the 66 features [AAC (20)+ PAAC (20)+ NAAC (20)+ PHYC (3)+ STRL (3)]. To compute the information gain, the ***InfoGainAttributeEval*** function available in ***RWeka*** package was used.

DCDNC: Software for discrimination of coding sequences (CDS) from non-coding sequences (Introns)

Each CDS and intron sequence was transformed into a numeric vector (of length five) based on five different indices i.e., Nucleotide frequencies by triplet sites (Nuft), Di-nucleotide frequencies by triplet sites (Dnft), Differential methylation intensity (Dmi), Triplet avoidance index (Tai) and Polypurine and polypyrimidine index (Popi). For computing the values of these indices we have written code in R-programming language.

ir-HSP: Improved Recognition of Heat Shock Proteins, Families and Sub-types

Four kinds of di-peptide compositions, *i.e.*, 0-spaced, 1-spaced, 2-spaced, and 3-spaced were used, which are nothing but the frequencies of all pairs of amino acids conditioned with 0, 1, 2, and 3 skips, respectively. Besides, all possible combinations of 0-, 1-, 2-, and 3-gap (spaced) amino acid pair compositions (GPC) were also used as features. Since, composition-transition-distribution (CTD), autocorrelation function (ACF), and pseudo-AAC (PAAC) features also take into account the local ordering of amino acids as similar to GPC, they were considered as features. For computing these features, ***BioSeqClass*** package of ***R***-software was used.

Overview on R and R-Studio
(Basic Commands and Data Handling)

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. R is a vehicle for newly developing methods of interactive data analysis. It has developed rapidly, and has been extended by a large collection of packages.

R environment

The R environment provides an integrated suite of software facilities for data manipulation, calculation and graphical display. It has

- a data handling and storage facility,
- a suite of operators for calculations on arrays and matrices,
- a large, integrated collection of intermediate tools for data analysis,
- graphical facilities for data analysis and display, and
- a well-developed, simple and effective programming language (called 'S') which includes conditionals, loops, user defined functions and input and output facilities.

Origin

R can be regarded as an implementation of the S language which was developed at Bell Laboratories by Rick Becker, John Chambers and Allan Wilks, and also forms the basis of the S-Plus systems. Robert Gentleman and Ross Ihaka of the Statistics Department of the University of Auckland started the project on R in 1995 and hence the name software has been named as 'R'.

R was introduced as an environment within which many classical and modern statistical techniques can be implemented. A few of these are built into the base R environment, but many are supplied as packages. There are a number of packages supplied with R (called "standard" and "recommended" packages) and many more are available through the CRAN family of Internet sites (via http://cran.r-project.org) and elsewhere.

Availability

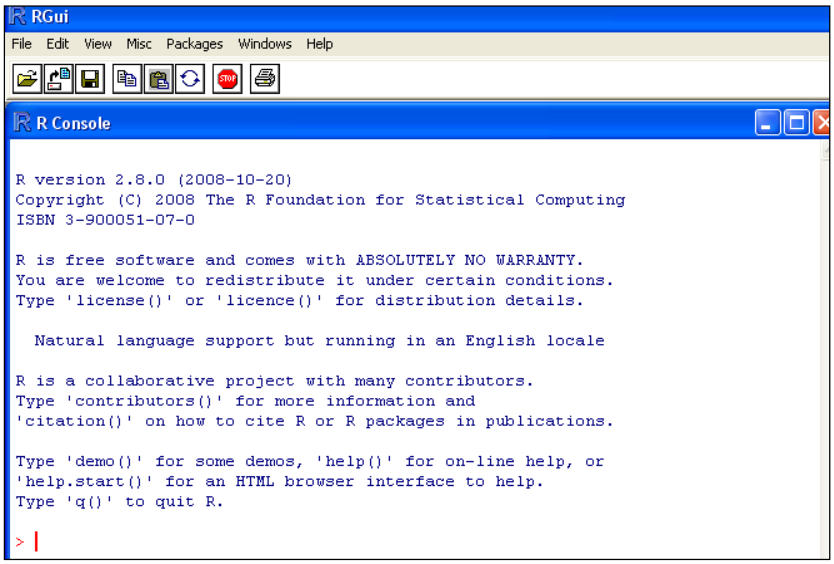
Since R is an open source project, it can be obtained freely from the website www.r-project.org. One can download R from any CRAN mirror out of several CRAN (Comprehensive R Archive Network) mirrors. Latest available version of R is *R version 3.6.1* and it has been released on 05.07.2019.

Download and Installation of R

Precompiled binary distributions of the base system and contributed packages, Windows and Mac users most likely want one of these versions of R: Linux , MacOS X, Windows.

Download and Installation of R for Windows is as follows:

- Visit http://cran.us.r-project.org/
- Browse Windows
- Click on "base" link - Binaries for base distribution (managed by Duncan Murdoch)
- Click "README.R-2.8.0" for Installation and other instructions
- Click "R-2.8.0-win32.exe" for downloading R-2.8.0 software
- Once download is complete, run "R-2.8.0-win32.exe".
- Follow the instructions to install R software.
- Click "R" shortcut icon. A "RGui" based "R Console" will appear



An exhaustive help in R can be seen by typing "help.start()".

Usage

R can work under Windows, UNIX and Mac OS. In this note, we consider usage of R in Windows set up only.

Important R-Packages

Basics

attribute: Data Attributes

- chron: Dates and Times
- classes: Data Types (not OO)

    - NA: Missing Values
    - category: Categorical Data
    - character: Character Data ("String") Operations
    - complex: Complex Numbers
- data: Environments, Scoping, Packages
- datasets: Datasets available by data()
- list: Lists
- manip: Data Manipulation
- package: Package Summaries
- sysdata: Basic System Variables

Graphics

- aplot: Add to Existing Plot / internal plot
- color: Color, Palettes etc
- device: Graphical Devices
- dplot: Computations Related to Plotting
- dynamic: Dynamic Graphics
- hplot: High-Level Plots
- iplot: Interacting with Plots

MASS (the book) uses

- classif: Classification
- neural: Neural Networks
- spatial: Spatial Statistics

Mathematics

- arith: Basic Arithmetic and Sorting
- array: Matrices and Arrays
    - algebra: Linear Algebra
- graphs: Graphs (not graphics), i.e nodes&edges, e.g. dendrograms
- logic: Logical Operators
- math: Mathematical Calculus etc
- optimize: Optimization
- symbolmath: "Symbolic Math", as polynomials, fractions

Programming, Input/Ouput, and Miscellaneous

- IO: Input/output
    - connection: Input/Output Connections
    - database: Interfaces to databases

    - file: Input/Output Files
- debugging: Debugging Tools
- documentation: Documentation
- environment: Session Environment
- error: Error Handling
- internal: Internal Objects (not part of API)
- iteration: Looping and Iteration
- methods: Methods and Generic Functions
- misc: Miscellaneous
- print: Printing
- programming: Programming
    - interface: Interfaces to Other Languages
- utilities: Utilities

Statistics

- cluster: Clustering
- datagen: Functions for generating data sets
- design: Designed Experiments
- distribution: Probability Distributions and Random Numbers
- htest: Statistical Inference
- models: Statistical Models
    - regression: Regression
        - nonlinear: Non-linear Regression
- multivariate: Multivariate Techniques
- nonparametric: Nonparametric Statistics
- robust: Robust/Resistant Techniques
- smooth: Curve (and Surface) Smoothing
    - loess: Loess Objects
- survey: Complex survey samples
- survival: Survival Analysis
- tree: Regression and Classification Trees
- ts: Time Series
- univar: simple univariate statistics

Difference with other packages

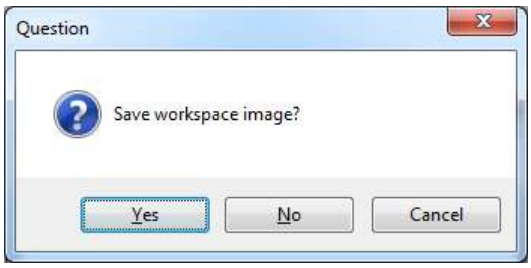
There is an important difference between R and the other statistical packages. In R, a statistical analysis is normally done as a series of steps, with intermediate results being stored in objects. Thus whereas SAS and SPSS will give large amount of output from a given

analysis, R will give minimal output and store the results in an object for subsequent interrogation by further R functions.

Invoking R

If properly installed, usually R has a shortcut icon on the desktop screen and/or you can find it under Start|All Programs|R menu.

To quit R, type *q()* at the R prompt (>) and press Enter key. A dialog box will ask whether to save the objects you have created during the session so that they will become available next time when R will be invoked.



R commands

i. R commands are case sensitive, so X and x are different symbols and would refer to different variables.

ii. Elementary commands consist of either expressions or assignments.

iii. If an expression is given as a command, it is evaluated, printed and the value is lost.

iv. An assignment also evaluates an expression and passes the value to a variable but the result is not automatically printed.

v. Commands are separated either by a semi-colon (';'), or by a newline.

vi. Elementary commands can be grouped together into one compound expression by braces '{' and '}'.

vii. Comments can be put almost anywhere, starting with a hashmark ('#'). Anything written after # marks to the end of the line is considered as a comment.

viii. Window can be cleared of lines by pressing Ctrl + L keys.

Executing commands from or diverting output to a file

If commands are stored in an external file, say 'D:/commands.txt' they may be executed at any time in an R session with the command

> source("d:/commands.txt")

For Windows Source is also available on the File menu.

The function *sink()*,

> sink("d:/record.txt")

will divert all subsequent output from the console to an external file, 'record.txt' in D drive. The command

> sink()

restores it to the console once again.

Simple manipulations of numbers and vectors

R operates on named data structures. The simplest such structure is the numeric vector, which is a single entity consisting of an ordered collection of numbers. To set up a vector named x, say, consisting of five numbers, namely 10.4, 5.6, 3.1, 6.4 and 21.7, use the R command

> x <- c(10.4, 5.6, 3.1, 6.4, 21.7)

The function *c()* assigns the five numbers to the vector x. The assignment operator (<-) 'points' to the object receiving the value of the expression. Once can use the '=' operator as an alternative.

A single number is taken as a vector of length one.

Assignments can also be made in the other direction, using the obvious change in the assignment operator. So the same assignment could be made using

> c(10.4, 5.6, 3.1, 6.4, 21.7) -> x

If an expression is used as a complete command, the value is printed. So now if we were to use the command

```
> 1/x
```

the reciprocals of the five values would be printed at the terminal.

The elementary arithmetic operators

+ addition
– subtraction
* multiplication
/ division
^ exponentiation

Arithmetic functions

log, exp, sin, cos, tan, sqrt,

Other basic functions

max(x) – maximum element of vector x,
min(x)- minimum element of vector x,
range (x) – range of the values of vector x ,
length(x) - the number of elements in x,
sum(x) - the total of the elements in x,
prod(x) – product of the elements in x
mean(x) – average of the elements of x
var(x) – sample variance of the elements of (x)
sort(x) – returns a vector with elements sorted in increasing order.

Logical operators

< - less than
<= less than or equal to
> greater than
>= greater than or equal to
== equal to
!= not equal to.

Other objects in R

Matrices or arrays - multi-dimensional generalizations of vectors.

Lists - a general form of vector in which the various elements need not be of the same type, and are often themselves vectors or lists.

Functions - objects in R which can be stored in the project's workspace. This provides a simple and convenient way to extend R.

Matrix facilities

A matrix is just an array with two subscripts. R provides many operators and functions those are available only for matrices. Some of the important R functions for matrices are

t(A) – transpose of the matrix A
nrow(A) – number of rows in the matrix A
ncol(A) – number of columns in the matrix A
A%*% B– Cross product of two matrices A and B
A*B – element by element product of two matrices A and B
diag (A) – gives a vector of diagonal elements of the square matrix A
diag(a) – gives a matrix with diagonal elements as the elements of vector a
eigen(A) – gives eigen values and eigen vectors of a symmetric matrix A
rbind (A,B) – concatenates two matrix A and B by appending B matrix below A matrix (They should have same number of columns)
cbind(A, B) - concatenates two matrix A and B by appending B matrix in the right of A matrix (They should have same number of rows)

Data frame

Data frame is an array consisting of columns of various mode (numeric, character, etc). Small to moderate size data frame can be constructed by *data.frame()* function. For example, following is an illustration how to construct a data frame from the car data*:

Make	Model	Cylinder	Weight	Mileage	Type
Honda	Civic	V4	2170	33	Sporty
Chevrolet	Beretta	V4	2655	26	Compact
Ford	Escort	V4	2345	33	Small
Eagle	Summit	V4	2560	33	Small
Volkswagen	Jetta	V4	2330	26	Small
Buick	Le Sabre	V6	3325	23	Large
Mitsubishi	Galant	V4	2745	25	Compact
Dodge	Grand Caravan	V6	3735	18	Van
Chrysler	New Yorker	V6	3450	22	Medium
Acura	Legend	V6	3265	20	Medium

```
> Make<-c("Honda","Chevrolet","Ford","Eagle","Volkswagen","Buick","Mitsbusihi",
+ "Dodge","Chrysler","Acura")
> Model=c("Civic","Beretta","Escort","Summit","Jetta","Le Sabre","Galant",
+ "Grand Caravan","New Yorker","Legend")
```

Note that the plus sign (+) in the above commands are automatically inserted when the carriage return is pressed without completing the list. Save some typing by using *rep()* command. For example, *rep("V4",5)* instructs R to repeat V4 five times.

```
> Cylinder<-c(rep("V4",5),"V6","V4",rep("V6",3))
> Cylinder
 [1] "V4" "V4" "V4" "V4" "V4" "V6" "V4" "V6" "V6" "V6"
> Weight<-c(2170,2655,2345,2560,2330,3325,2745,3735,3450,3265)
> Mileage<-c(33,26,33,33,26,23,25,18,22,20)
> Type<-c("Sporty","Compact",rep("Small",3),"Large","Compact","Van",rep("Medium",2))
```

Now *data.frame()* function combines the six vectors into a single data frame.

```
> Car<-data.frame(Make,Model,Cylinder,Weight,Mileage,Type)
> Car
         Make          Model Cylinder Weight Mileage    Type
1       Honda          Civic       V4   2170      33  Sporty
2   Chevrolet        Beretta       V4   2655      26 Compact
3        Ford         Escort       V4   2345      33   Small
4       Eagle         Summit       V4   2560      33   Small
5  Volkswagen          Jetta       V4   2330      26   Small
6       Buick       Le Sabre       V6   3325      23   Large
7  Mitsbusihi         Galant       V4   2745      25 Compact
8       Dodge  Grand Caravan       V6   3735      18     Van
9    Chrysler     New Yorker       V6   3450      22  Medium
10      Acura         Legend       V6   3265      20  Medium

> names(Car)
[1] "Make"     "Model"    "Cylinder" "Weight"   "Mileage"  "Type"
```

Just as in matrix objects, partial information can be easily extracted from the data frame:

```
> Car[1,]

   Make Model Cylinder Weight Mileage   Type

1 Honda Civic       V4   2170      33 Sporty
```

In addition, individual columns can be referenced by their labels:

```
> Car$Mileage
 [1] 33 26 33 33 26 23 25 18 22 20
> Car[,5]        #equivalent expression
> mean(Car$Mileage)    #average mileage of the 10 vehicles
[1] 25.9
> min(Car$Weight)
[1] 2170
```

table() command gives a frequency table:

```
> table(Car$Type)

Compact   Large  Medium   Small  Sporty     Van
      2       1       2       3       1       1
```

If the proportion is desired, type the following command instead:

```
> table(Car$Type)/10

Compact   Large  Medium   Small  Sporty     Van
    0.2     0.1     0.2     0.3     0.1     0.1
```

Note that the values were divided by 10 because there are that many vehicles in total. If you don't want to count them each time, the following does the trick:

```
> table(Car$Type)/length(Car$Type)
```

Cross tabulation is very easy, too:

```
> table(Car$Make, Car$Type)

            Compact Large Medium Small Sporty Van
  Acura       0       0      1      0      0      0
  Buick       0       1      0      0      0      0
  Chevrolet   1       0      0      0      0      0
  Chrysler    0       0      1      0      0      0
  Dodge       0       0      0      0      0      1
  Eagle       0       0      0      1      0      0
  Ford        0       0      0      1      0      0
  Honda       0       0      0      0      1      0
  Mitsbusihi  1       0      0      0      0      0
  Volkswagen  0       0      0      1      0      0
```

What if you want to arrange the data set by vehicle weight? *order()* gets the job done.

```
> i<-order(Car$Weight);i
 [1] 1 5 3 4 2 7 10 6 9 8
> Car[i,]
         Make          Model Cylinder Weight Mileage    Type
1       Honda          Civic       V4   2170      33  Sporty
5  Volkswagen          Jetta       V4   2330      26   Small
3        Ford         Escort       V4   2345      33   Small
4       Eagle         Summit       V4   2560      33   Small
2   Chevrolet        Beretta       V4   2655      26 Compact
7  Mitsbusihi         Galant       V4   2745      25 Compact
10      Acura         Legend       V6   3265      20  Medium
6       Buick      Le Sabre       V6   3325      23   Large
```

```
9     Chrysler     New Yorker        V6    3450       22   Medium
8        Dodge Grand Caravan        V6    3735       18      Van
```

Creating/editing data objects

```
> y<-c(1,2,3,4,5);y
[1] 1 2 3 4 5
```

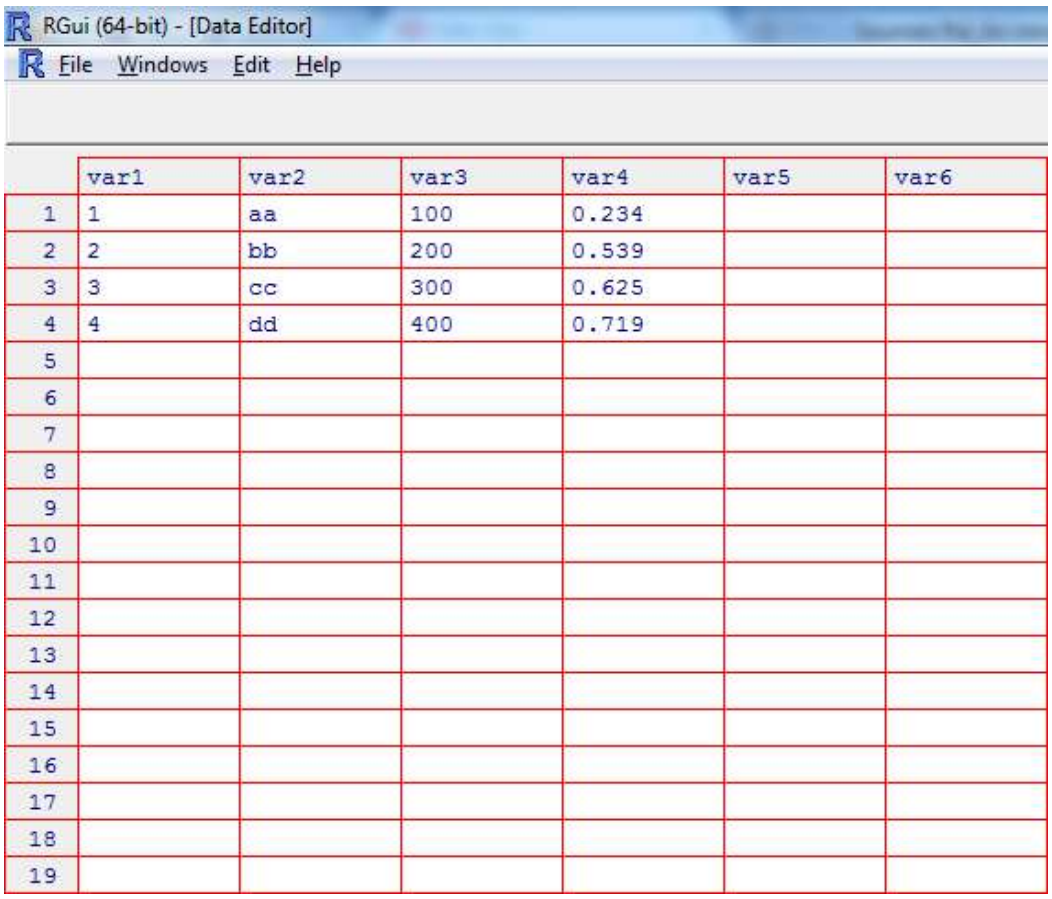
If you want to modify the data object, use *edit()* function and assign it to an object. For example, the following command opens R Editor for editing.

```
> y<-edit(y)
```

If you prefer entering the data.frame in a spreadsheet style data editor, the following command invokes the built-in editor with an empty spreadsheet.

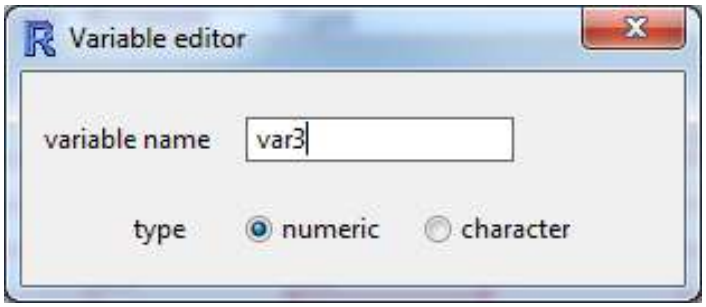
```
> data1<-edit(data.frame())
```

After entering a few data points, it looks like this:



	var1	var2	var3	var4	var5	var6
1	1	aa	100	0.234		
2	2	bb	200	0.539		
3	3	cc	300	0.625		
4	4	dd	400	0.719		
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						

You can also change the variable name by clicking once on the cell containing it. Doing so opens a dialog box:



When finished, click  in the upper right corner of the dialog box to return to the Data Editor window. Close the Data Editor to return to the R command window (R Console). Check the result by typing:

```
> data1
```

Reading data from files

When data files are large, it is better to read data from external files rather than entering data through the keyboard. To read data from an external file directly, the external file should be arranged properly.

The first line of the file should have a name for each variable. Each additional line of the file has the values for each variable.

Input file form with names and row labels:

Price	Floor	Area	Rooms	Age	is New
52.00	111.0	830	5	6.2	no
54.75	128.0	710	5	7.5	no
57.50	101.0	1000	5	4.2	yes
57.50	131.0	690	6	8.8	no
59.75	93.0	900	5	1.9	yes

...

By default numeric items (except row labels) are read as numeric variables and non-numeric variables, such as isNew in the example, as factors. This can be changed if necessary.

The function *read.table()* can then be used to read the data frame directly

> HousePrice <- read.table("d:/houses.data", header = TRUE)

Reading comma delimited data

The following commands can be used for reading comma delimited data into R.

read.csv(filename) This command reads a .CSV file into R. You need to specify the exact filename with path.

read.csv(file.choose()) This command reads a .CSV file but the *file.choose()* part opens up an explorer type window that allows you to select a file from your computer. By default, R will take the first row as the variable names.

read.csv(file.choose(), header=T)

This reads a .CSV file, allowing you to select the file, the header is set explicitly. If you change to header=F then the first row will be treated like the rest of the data and not as a label.

Storing variable names

Through *read.csv()* or *read.table()* functions, data along with variable labels is read into R memory. However, to read the variables' names directly into R, one should use *attach(dataset)* function. For example,

>attach(HousePrice)

causes R to directly read all the variables' names eg. Price, Floor, Area etc. it is a good practice to use the *attach(datafile)* function immediately after reading the *datafile* into R.

Packages

All R functions and datasets are stored in packages. The contents of a package are available only when the package is loaded. This is done to run the codes efficiently without much memory usage. To see which packages are installed at your machine, use the command

> library()

To load a particular package, use a command like

> library(forecast)

Users connected to the Internet can use the *install.packages()* and *update.packages()* functions to install and update packages. Use *search()* to display the list of packages that are loaded.

Standard packages

The standard (or base) packages are considered part of the R source code. They contain the basic functions those allow R to work with the datasets and standard statistical and graphical functions. They should be automatically available in any R installation.

Contributed packages and CRAN

There are a number of contributed packages for R, written by many authors. Various packages deal with various analyses. Most of the packages are available for download from CRAN (https://cran.r-project.org/web/packages/), and other repositories such as Bioconductor (http://www.bioconductor.org/). The collection of available packages changes frequently. As on June 07, 2019, the CRAN package repository contains 14346 available packages.

Getting Help

Complete help files in HTML and PDF forms are available in R. To get help on a particular command/function etc., type *help (command name)*. For example, to get help on function 'mean', type *help(mean)* as shown below

> help(mean)

This will open the help file with the page containing the description of the function mean.

Another way to get help is to use "?" followed by function name. For example,

>?mean

will open the same window again.

In this lecture note, all R commands and corresponding outputs are given in Courier New font to differentiate from the normal texts. Since R is case-sensitive, i.e. typing *Help(mean)*, would generate an error message,

> Help(mean)

Error in Help(mean) : could not find function "Help"

Further Readings

Various documents are available in https://cran.r-project.org/manuals.html from beginners' level to most advanced level. The following manuals are available in pdf form:

1. An Introduction to R
2. R Data Import/Export

3. R Installation and Administration
4. Writing R Extensions
5. The R language definition
6. R Internals
7. The R Reference Index

RStudio

RStudio is an integrated development environment (IDE) that allows to interact with R more readily. RStudio is similar to the standard RGui, but is considerably more user friendly. It has more drop-down menus, windows with multiple tabs, and many customization options.

Installation of RStudio

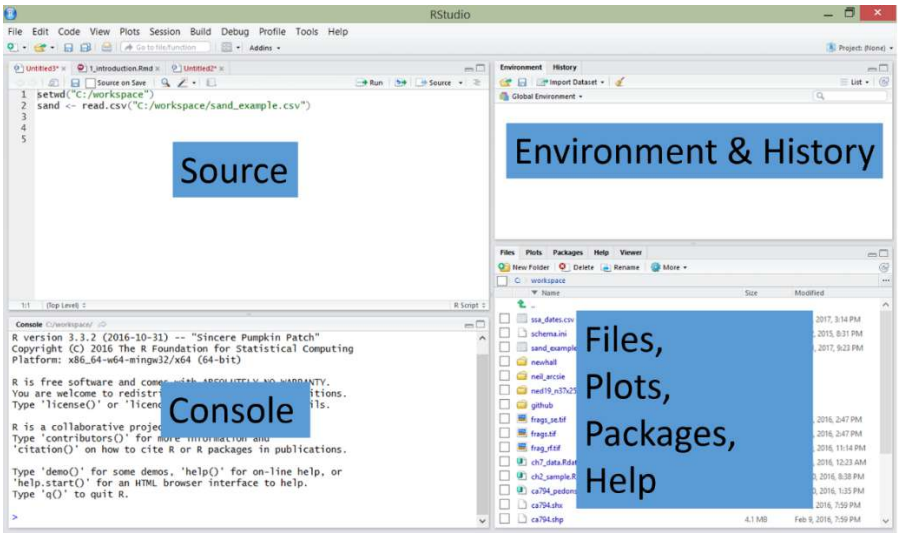
RStudio requires R 3.0.1+ that means R software should be pre-installed before using RStudio.

RStudio 1.2 requires a 64-bit operating system, and works exclusively with the 64 bit version of R. If you are on a 32 bit system or need the 32 bit version of R, you can use an older version of RStudio (https://support.rstudio.com/hc/en-us/articles/206569407-Older-Versions-of-RStudio).

RStudio free desktop version can be downloaded from the following link:

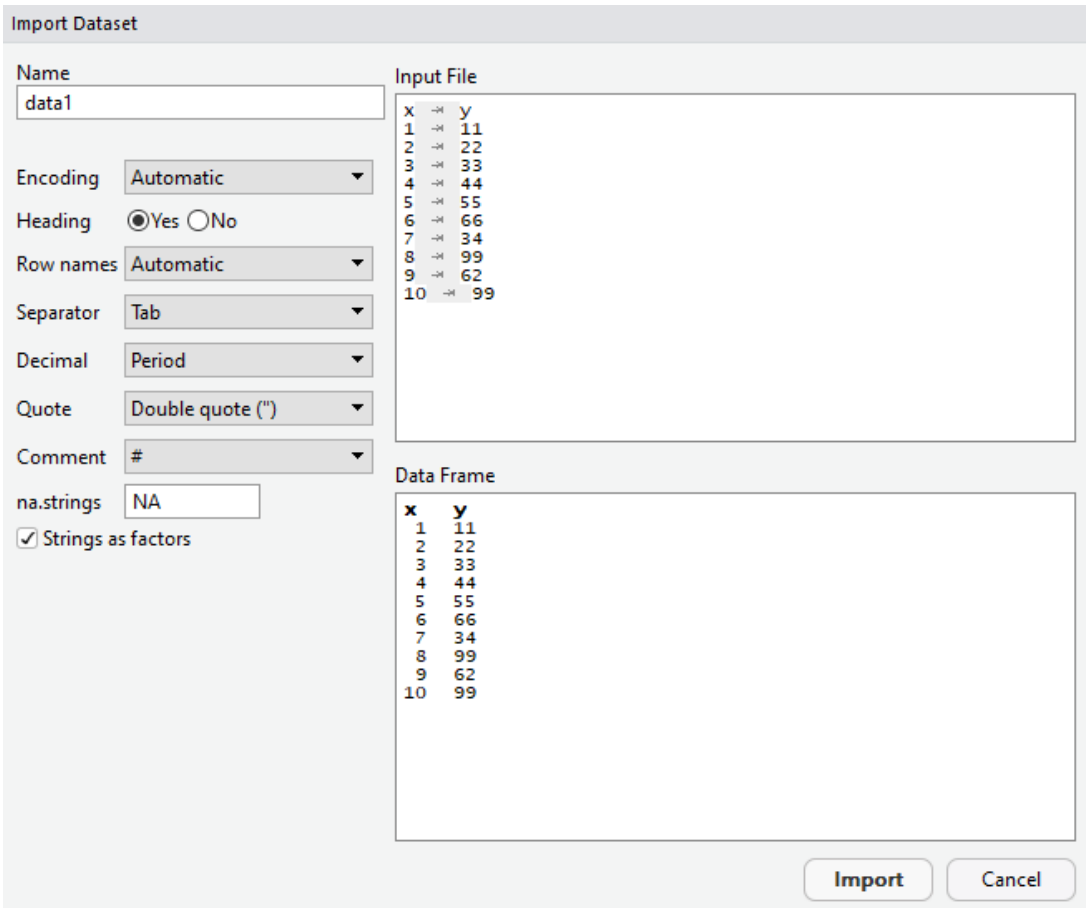
https://www.rstudio.com/products/rstudio/download/#download

The first time RStudio is opened, three windows are seen. A forth window is hidden by default, but can be opened by clicking the **File** drop-down menu, then **New File**, and then **R Script**.



Importing Data in R Studio

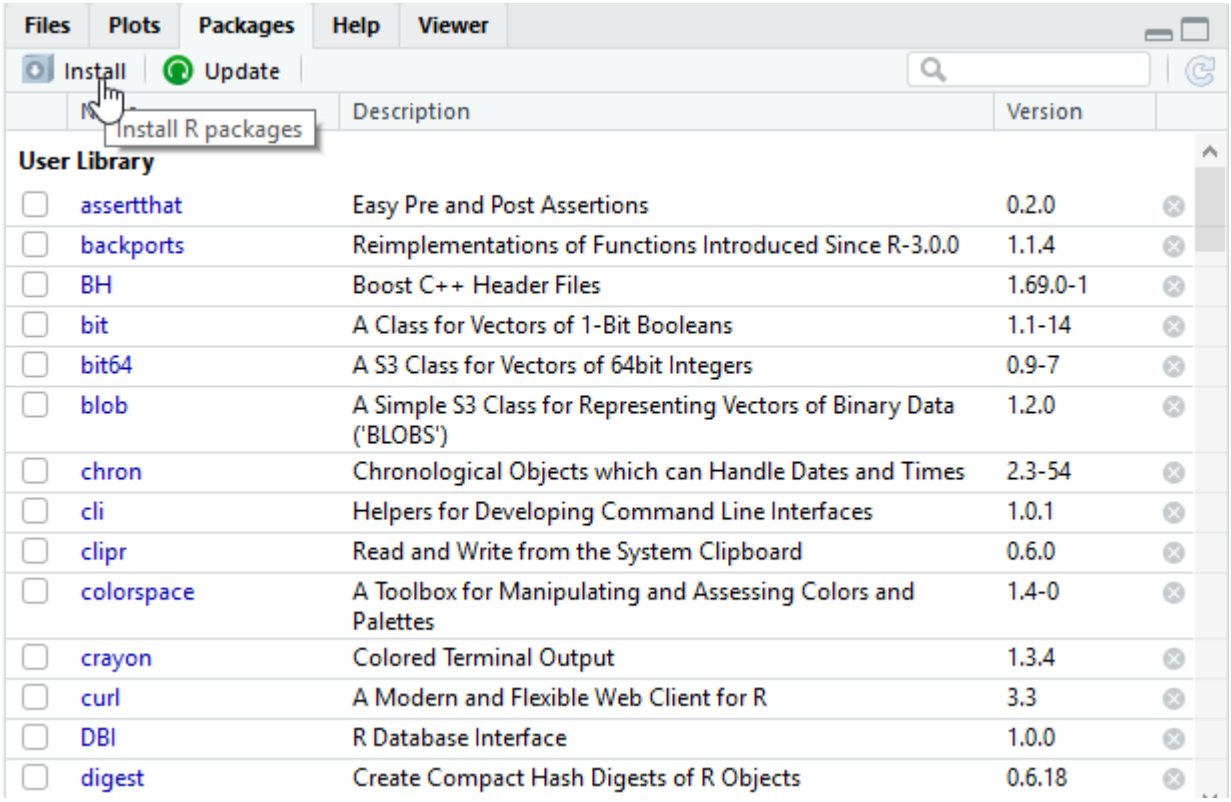
1. Click on the import dataset button in the top-right section under the environment tab. Select the file you want to import and then click open. The Import Dataset dialog will appear as shown below

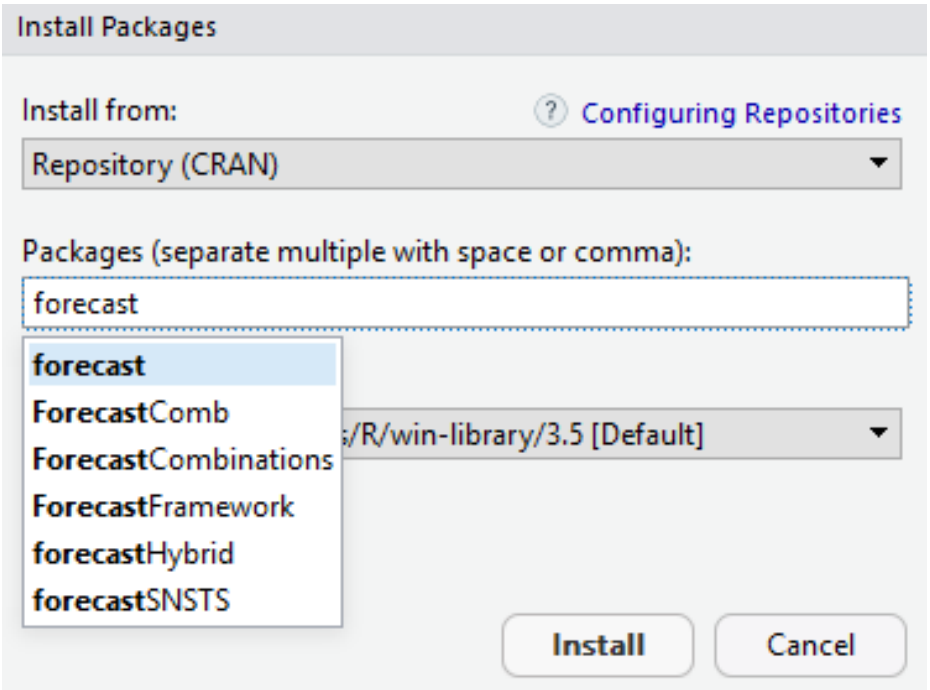


2. After setting up the preferences of separator, name and other parameters, click on the Import button. The dataset will be imported in R Studio and assigned to the variable name as set before.

Installing Packages in RStudio

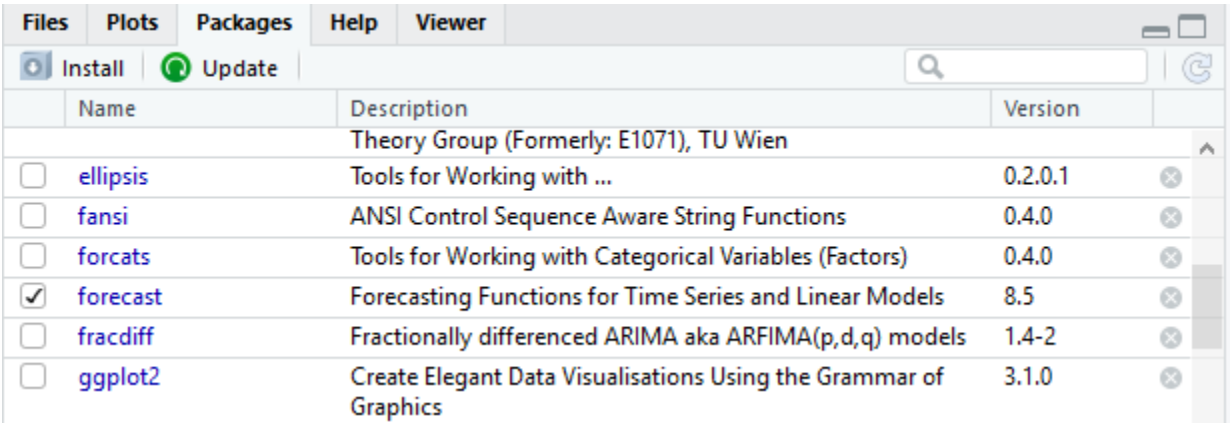
Within the **Packages** tab, a list of all the packages currently installed on the working computer and 2 buttons labeled either “Install” or “Update” are seen. To install a new package simply select the Install button. It is possible to install one or more than one packages at a time by simply separating them with a comma.





Loading Packages in RStudio

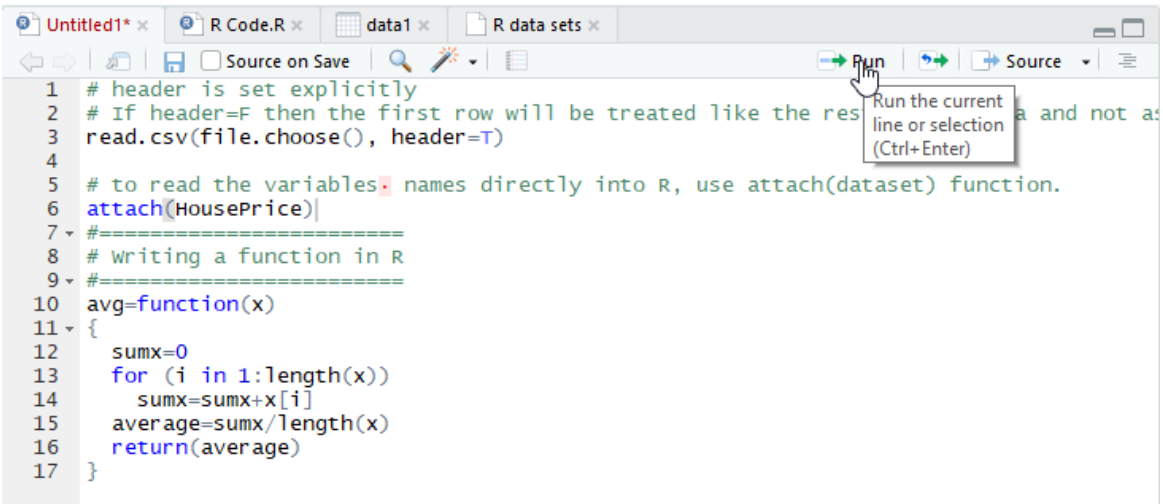
Once a package is installed, it must be loaded into the R session to be used.



Writing Scripts in RStudio

RStudio's Source Tabs serve as a built-in text editor. Prior to executing R functions at the Console, commands are typically written down (or scripted). To write a script, simply open a new R script file by clicking File>New File>R Script. Within the text editor type out a sequence of functions.

- Place each function (e.g. read.csv()) on a separate line.
- If a function has a long list of arguments, place each argument on a separate line.
- A command can be executed from the text editor by placing the cursor on a line and typing Crtl + Enter, or by clicking the Run button.
- An entire R script file can be executed by clicking the Source button.

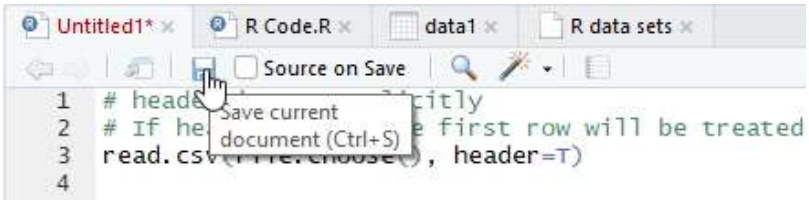


Saving R files in RStudio

In R, several types of files can be saved to keep track of the work performed. The file types include: script, workspace, history and graphics.

R script (.R)

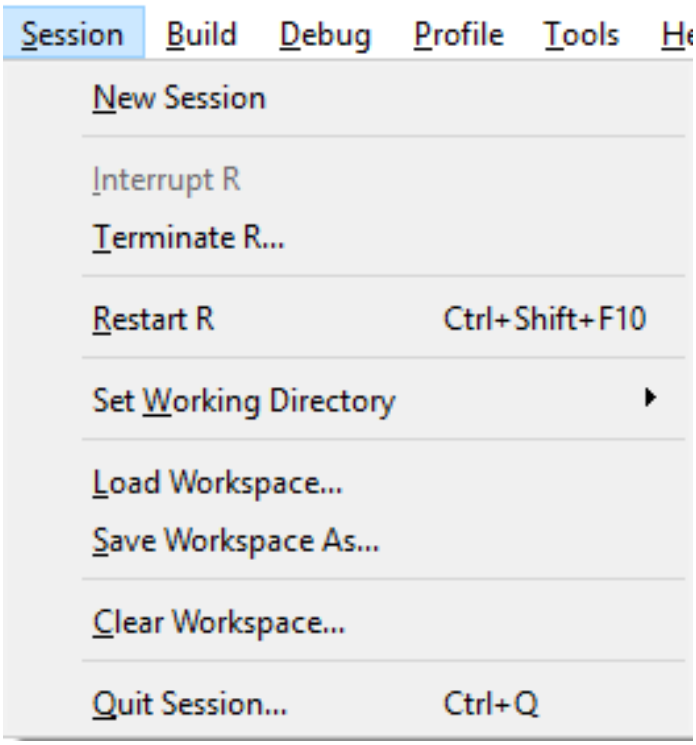
An R script is a text file of R commands that have been typed. To save R scripts in RStudio, click the save button from R script tab. Save scripts with the .R extension.



To open an R script, click the file icon.

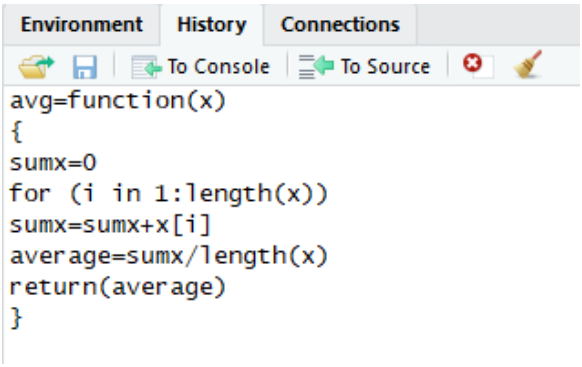
Workspace (.Rdata)

The R workspace consists of all the data objects created or loaded during the R session. It is possible to save or load the workspace at any time during the R session from the menu by clicking Session>Save Workspace As.., or the save button on the Environment Tab.



R history (.Rhistory)

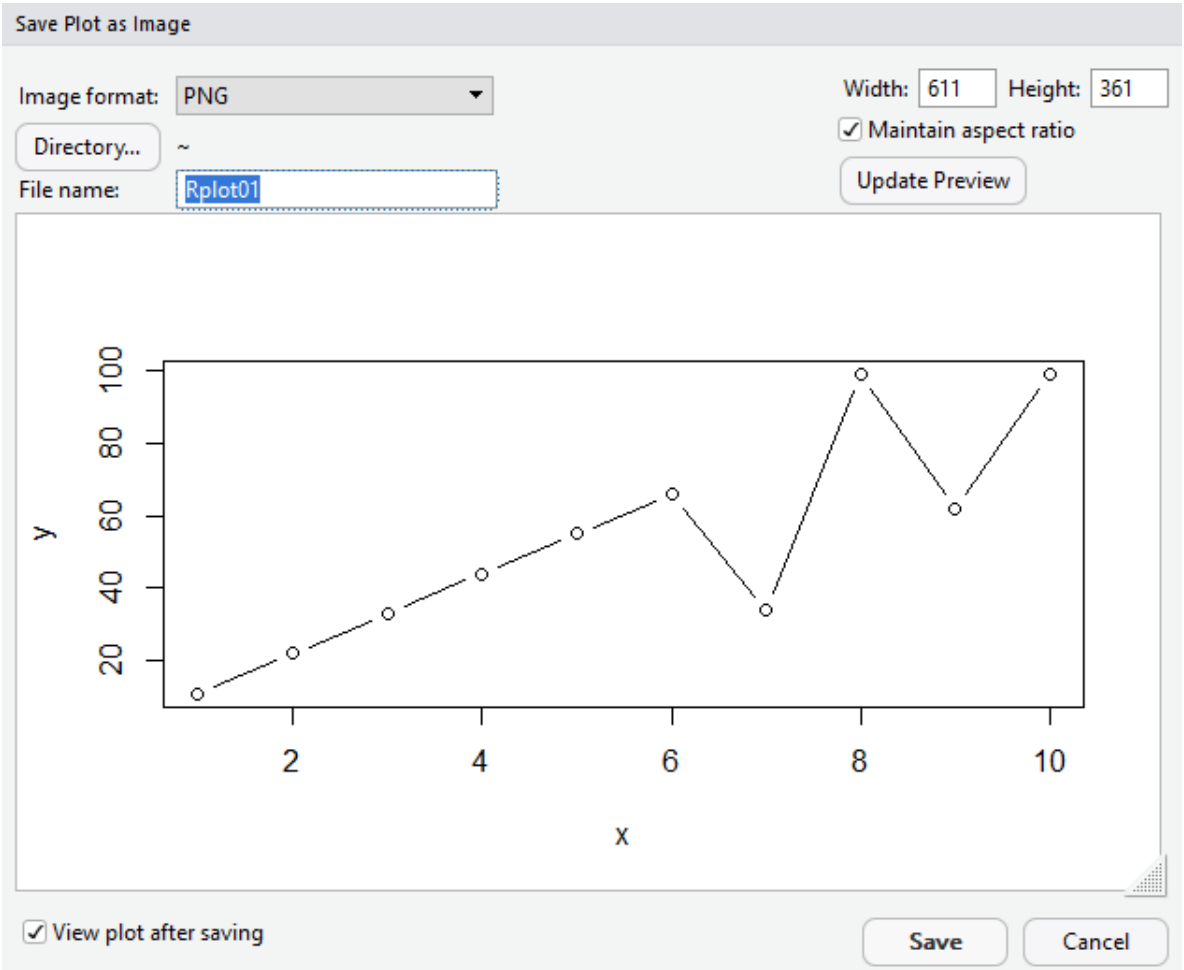
Rhistory file is a text file that lists all of the commands that have been executed. It does not keep a record of the results. To load or save R history from the History Tab click the **Open File** or **Save** button.



R Graphics

Graphic outputs can be saved in various formats like pdf, png, jpeg, bmp etc.

To save a graphic: (1) Click the **Plots** Tab window, (2) click the **Export** button, (3) **Choose** desired format, (4) **Modify** the export settings as desired and (4) click **Save**.



References

R Development Core Team (2008). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. http://www.R-project.org.

http://ncss-tech.github.io/stats_for_soil_survey/chapters/1_introduction/1_introduction.html

http://web.cs.ucla.edu/~gulzar/rstudio/basic-tutorial.html

http://www.gardenersown.co.uk/Education/Lectures/R/index.htm

https://www.cran.r-project.org

https://www.rstudio.com/

Matloff, N. (2011). *The art of R programming: A tour of statistical software design*. No Starch Press.

Venables, W. N., Smith, D. M. and R Development Core Team (2009). An introduction to R: Notes on R: A programming Environment for Data Analysis and Graphics, version 1.7. 1.

Descriptive Statistics and Linear Models Using R

Descriptive statistics are used to describe the basic features of the data in a study. They provide summaries about the sample and the measures. Together with simple graphics analysis, they form the basis of quantitative analysis of data. Descriptive statistics can be divided into 2 categories:

Measures of Central Tendency

Central tendency or measure of central tendency is a central or typical value for a probability distribution. The most common measures of central tendency are the arithmetic mean, the median and the mode.

Mean: The mean is equal to the sum of all the values in the data set divided by the number of values in the data set. So, if we have n values in a data set and they have values x_1, x_2, ..., x_n, the sample mean, usually denoted by $\bar{x}$, is:

$$\bar{x} = \frac{(x_1 + x_2 + \cdots + x_n)}{n}$$

Or $$\bar{x} = \frac{\sum x}{n}$$

Median: The median is the middle value for a set of data that has been arranged in order of magnitude. To find the median, order the data from smallest to largest, and then find the data point that has an equal amount of values above it and below it. The method for locating the median varies depending on whether the dataset has an even or odd number of values. If the dataset is having odd number of observations, there will be only one middlemost observation and the corresponding value is the median. When there is an even number of values, it is required to count in to the two innermost values and then take the average.

Mode: The mode is the value that occurs most frequently in the data set. If the data have multiple values that are tied for occurring most frequently, there exists a multimodal distribution. If no value repeats, the data do not have a mode. Typically, the mode can be used with categorical, ordinal, and discrete data. In fact, the mode is the only measure of central tendency that can be used with categorical data such as the most preferred way of transport out of car, train and bus.

Measures of Dispersion

The measure of dispersion shows the scatterings of the data. It shows the homogeneity or the

heterogeneity of the distribution of the observations. The measure of dispersion is categorized as:

(a) **Absolute measure of dispersion**:

(i) The measures which express the scattering of observation in terms of distances i.e., range, quartile deviation.

(ii) The measure which expresses the variations in terms of the average of deviations of observations like mean deviation and standard deviation.

(b) **Relative measure of dispersion**: Relative measure of dispersion is used for comparing distributions of two or more data set and for unit free comparison. They are coefficient of range, coefficient of mean deviation, coefficient of quartile deviation, coefficient of variation, and coefficient of standard deviation.

Range: It is the difference between two extreme observations of the data set. If X_{max} and X_{min} are the two extreme observations then

Range = $X_{max} - X_{min}$

Quartile Deviation: The quartiles divide a data set into quarters. The first quartile, (Q_1) is the middle number between the smallest number and the median of the data. The second quartile, (Q_2) is the median of the data set. The third quartile, (Q_3) is the middle number between the median and the largest number.

Quartile deviation or semi-inter-quartile deviation is

$Q = ½ \times (Q_3 - Q_1)$

Mean Deviation: Mean deviation is the arithmetic mean of the absolute deviations of the observations from a measure of central tendency. If $x_1, x_2, \ldots, x_n$ are the set of observation, then the mean deviation of x about the average A (mean, median, or mode) is

Mean deviation from average $A = 1/n \, [\sum_i |x_i - A|]$

For a grouped frequency, it is calculated as:

Mean deviation from average $A = 1/N \, [\sum_i f_i \, |x_i - A|]$, $N = \sum f_i$

Here, x_i and f_i are respectively the mid value and the frequency of the i^{th} class interval.

Standard Deviation: A standard deviation is the positive square root of the arithmetic mean of the squares of the deviations of the given values from their arithmetic mean. It is also referred to as root mean square deviation. The standard deviation is given as

$\sigma = [(\Sigma_i (y_i - \bar{y})^2 / n]\ ½ = [(\Sigma_i y_i^2 / n) - \bar{y}^2]\ ½$

For a grouped frequency distribution, it is

$\sigma = [(\Sigma_i f_i (y_i - \bar{y})^2 / n]\ ½ = [(\Sigma_i f_i y_i^2 / n) - \bar{y}^2]\ ½$

The square of the standard deviation is the **variance**. It is also a measure of dispersion.

$\sigma^2 = [(\Sigma_i (y_i - \bar{y})^2 / n] = [(\Sigma_i y_i^2 / n) - \bar{y}^2]$

For a grouped frequency distribution, it is

$\sigma^2 = [(\Sigma_i f_i (y_i - \bar{y})^2 / n] = [(\Sigma_i f_i y_i^2 / n) - \bar{y}^2]$.

If instead of a mean, any other arbitrary number, say A, is chosen, the standard deviation becomes the root mean deviation.

Variance of the Combined Series: If σ_1, σ_2 are two standard deviations of two series of sizes n_1 and n_2 with means $\bar{y}_1$ and $\bar{y}_2$. The variance of the two series of sizes n1 + n2 is:

$\sigma^2 = (1/ n_1 + n_2) \div [n_1 (\sigma_1^2 + d_1^2) + n_2 (\sigma_2^2 + d_2^2)]$

where, $d_1 = \bar{y}_1 - \bar{y}$, $d_2 = \bar{y}_2 - \bar{y}$, and $\bar{y} = (n_1\bar{y}_1 + n_2\bar{y}_2) \div (n_1 + n_2)$.

Coefficient of Dispersion: Coefficient of dispersion is used to compare the variability of the two series which differ widely in their averages. Also, it is used when the unit of measurement is different. It is needed to calculate the coefficients of dispersion along with the measure of dispersion. The coefficients of dispersion (C.D.) based on different measures of dispersion are:

Based on Range = $(X_{max} - X_{min}) / (X_{max} + X_{min})$.

C.D. based on quartile deviation = $(Q_3 - Q_1) / (Q_3 + Q_1)$.

Based on mean deviation = Mean deviation/average from which it is calculated.

For Standard deviation = S.D. / Mean

Coefficient of Variation: This is 100 times the coefficient of dispersion based on standard deviation.

C.V. = 100 × (S.D. / Mean) = $(\sigma/\bar{y}) \times 100$.

Skewness and Kurtosis

The average and measure of dispersion can describe the distribution but they are not sufficient to describe the nature of the distribution. For this purpose, other concepts known as

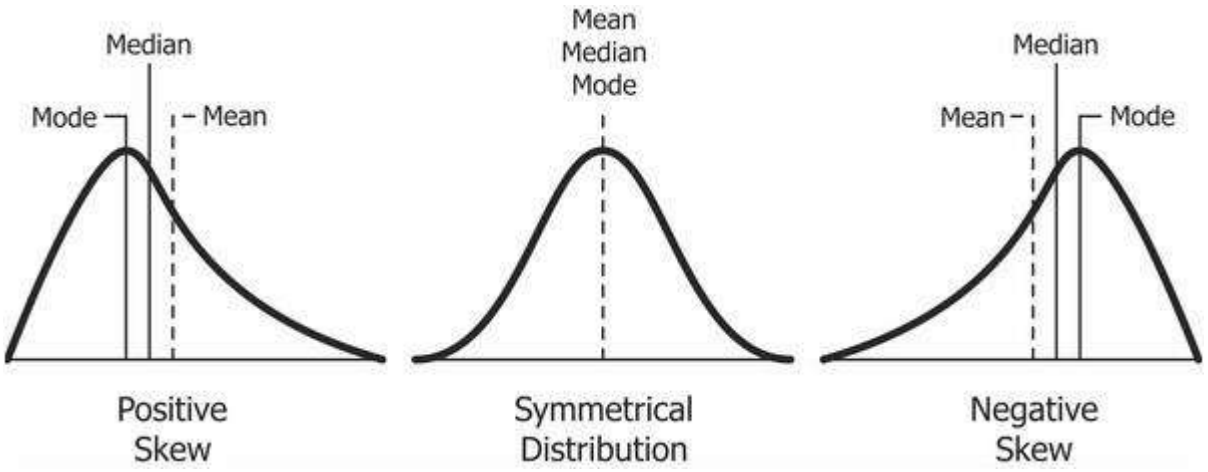
Skewness and Kurtosis, are used.

Skewness

Skewness is a measure of the asymmetry of the probability distribution of a real-valued random variable about its mean. The skewness value can be positive or negative, or undefined. In a perfect normal distribution, the tails on either side of the curve are exact mirror images of each other.

When a distribution is skewed to the left, the tail on the curve's left-hand side is longer than the tail on the right-hand side, and the mean is less than the mode. This situation is also called negative skewness.

When a distribution is skewed to the right, the tail on the curve's right-hand side is longer than the tail on the left-hand side, and the mean is greater than the mode. This situation is also called positive skewness.



To calculate skewness coefficient of the sample, there are two methods:

Pearson First Coefficient of Skewness (Mode skewness):

(Mean – Mode) / Standard Deviation

Pearson Second Coefficient of Skewness (Median skewness):

3(Mean – Median) / Standard Deviation

The other measure uses the β ('beta') coefficient which is given by, $\beta_1 = \frac{\mu_3^2}{\mu_2^3}$ where, μ_2 and μ_3 are the second and third central moments. The second central moment μ_2 is nothing but the variance. The sample estimate of this coefficient is $b_1 = \frac{m_3^2}{m_2^3}$ where m_2 and m_3 are the second and third sample central moments.

Interpretations

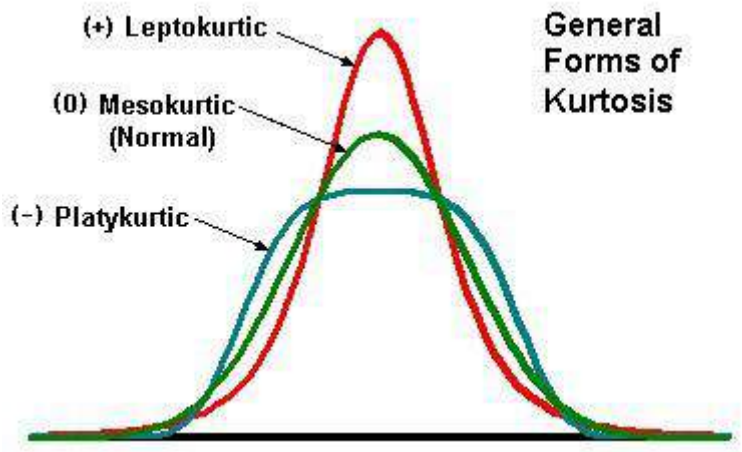
a. The direction of skewness is given by the sign. A zero means no skewness at all or the distribution is symmetric.

b. A negative value means the distribution is negatively skewed. A positive value means the distribution is positively skewed.

c. The coefficient compares the sample distribution with a normal distribution. The larger the value, the larger the distribution differs from a normal distribution.

Kurtosis

The measure of kurtosis is used to find out existence of outliers. Kurtosis is a measure of whether the data are heavy-tailed (profusion of outliers) or light-tailed (lack of outliers) relative to a normal distribution.



There are three types of Kurtosis:

a. Mesokurtic: This is the distribution which has similar kurtosis as normal distribution kurtosis, which is zero.

b. Leptokurtic: This is the distribution which has kurtosis greater than a Mesokurtic distribution. Tails of such distributions are thick and heavy. If the curve of a distribution is more peaked than Mesokurtic curve, it is referred to as a Leptokurtic curve.

c. Platykurtic: This is the distribution which has kurtosis lesser than a Mesokurtic distribution. Tails of such distributions are thinner. If a curve of a distribution is less peaked than a Mesokurtic curve, it is referred to as a Platykurtic curve.

Kurtosis is measured by Pearson's coefficient, β_2 ('beta - two'). It is given by $\beta_2 = \frac{\mu_4}{\mu_2^2}$.

The sample estimate of this coefficient is $b_2 = \frac{m_4}{m_2^2}$ where, m_4 is the fourth central moment.

The distribution is called normal if $b_2 = 3$. When b_2 is more than 3 the distribution is said to be leptokurtic. If b_2 is less than 3 the distribution is said to be platykurtic.

The main difference between skewness and kurtosis is that the skewness refers to the degree of symmetry, whereas the kurtosis refers to the degree of presence of outliers in the distribution.

Histogram

Histogram shows the frequency distribution of a quantitative variable as vertical bars with area of the bar denotes the frequency of items found in each class interval. Histograms are useful to assess the distribution of the variable.

Box Plot

A box plot or boxplot is a method for graphically depicting groups of numerical data through their quartiles. This is a standardized way of displaying the distribution of data based on a five number summary ("minimum", first quartile (Q1), median, third quartile (Q3), and "maximum").

Descriptive Statistics using R

In this section, a set of functions available in R are presented to describe and explore data.

```
# mean() is used to calculate mean value in a data series.

# mean(x, trim = 0, na.rm = FALSE, ...)

# Create a vector.

> x = c(12,7,3,4.2,18,2,54,-21,8,-5)
# Find Mean.

> result.mean = mean(x)

> print(result.mean)

[1] 8.22
```

When trim = 0.3, 3 values from each end will be dropped from the
calculations to find mean.

> result.mean = mean(x,trim = 0.3) # Find Mean.

> print(result.mean)

[1] 5.55

If there are missing values, then the mean function returns NA.

To drop the missing values from the calculation use na.rm = TRUE #which means remove the NA values.

Create a vector.

> x = c(12,7,3,4.2,18,2,54,-21,8,-5,NA)

Mean can't be calculated as x contains NA.
> result.mean = mean(x)

> print(result.mean)

[1] NA

Find mean after dropping NA values.

> result.mean = mean(x,na.rm = TRUE)

> print(result.mean)

[1] 8.22

The median() function is used in R to calculate median value.

function: median(x, na.rm = FALSE)

Create the vector.

> x = c(12,7,3,4.2,18,2,54,-21,8,-5)
Find the median.

```
> median.result = median(x)
> print(median.result)
[1] 5.6
# Find the mode.
# Create the vector.
> x = c(8,2,7,1,2,9,8,2,10,9)
> y = table(x)
> print(y)
x
 1  2  7  8  9 10
 1  3  1  2  2  1
> names(y)[which(y==max(y))]   # calculate mode
[1] "2"
# Testing if there are two or more numbers with same frequency.
> x = c(8,2,7,1,2,9,8,2,10,9,8)
> sort(x)
[1]  1  2  2  2  7  8  8  8  9  9 10
> names(table(x))[table(x)==max(table(x))]  # calculate mode
[1] "2" "8"
# When x is a character vector.
> x = c("o","it","the","it","it")
> sort(table(x))
x
  o the  it
  1   1   3
> names(table(x))[table(x)==max(table(x))]  # calculate mode
```

```
[1] "it"

#================================================
# Mean, Median and Mode using airquality dataset.
#================================================

> dim(airquality)        # 153 obs. of  6 variables.

[1] 153   6

# Column names with missing Values.

> names(airquality)[colSums(is.na(airquality)) > 0]

[1] "Ozone"   "Solar.R"

> x = airquality$Solar.R         # define x.

> table(is.na(x))

FALSE  TRUE

146     7

> mean(x)       # mean of x can't be calculated as x is having values NA.

[1] NA

> mean(x, na.rm = TRUE)      # mean of x after removing NA.

[1] 185.9315

> median(x, na.rm = TRUE)  # median of x after removing NA.

[1] 205

> names(table(x))[table(x)==max(table(x))]     # mode of x.

[1] "238" "259"

# produce result summaries.

> summary(airquality)                    # summary statistics.

    Ozone          Solar.R          Wind            Temp          Month

 Min.   :  1.00   Min.   :  7.0   Min.   : 1.700   Min.   :56.00   Min.   :5.000

 1st Qu.: 18.00   1st Qu.:115.8   1st Qu.: 7.400   1st Qu.:72.00   1st Qu.:6.000
```

```
 Median : 31.50   Median :205.0   Median : 9.700   Median :79.00   Median :7.000
 Mean   : 42.13   Mean   :185.9   Mean   : 9.958   Mean   :77.88   Mean   :6.993
 3rd Qu.: 63.25   3rd Qu.:258.8   3rd Qu.:11.500   3rd Qu.:85.00   3rd Qu.:8.000
 Max.   :168.00   Max.   :334.0   Max.   :20.700   Max.   :97.00   Max.   :9.000
 NA's   :37       NA's   :7
     Day
 Min.   : 1.0
 1st Qu.: 8.0
 Median :16.0
 Mean   :15.8
 3rd Qu.:23.0
 Max.   :31.0
#================================
# Variance and Standard Deviation.
#================================
```

Find the variance of eruption duration in the data set faithful.

```
> duration = faithful$eruptions
> var(duration)
[1] 1.3027
```

Find the standard deviation of the eruption duration in the data set # faithful.

```
> sd(duration) # Standard Deviation.
[1] 1.141371
```

Find the skewness of eruption duration in the data set faithful.

```
> install.packages("e1071")          # install package e1071.
> library(e1071)                     # load e1071
> duration = faithful$eruptions      # eruption durations
> skewness(duration)                 # apply the skewness function
```

[1] -0.41355

The skewness of eruption duration is -0.41355. It indicates that the eruption duration distribution is skewed towards the left.

Kurtosis.

```
> library(e1071)                  # load e1071
> duration = faithful$eruptions   # eruption durations
> kurtosis(duration)              # apply the kurtosis function
```

[1] -1.5116

The excess kurtosis of eruption duration is -1.5116, which indicates that eruption duration distribution is platykurtic.

Simulate 10000 samples from a normal distribution with mean 55, and # standard deviation 4.5, then compute and interpret the skewness and # kurtosis, and plot the histogram and boxplot.

#Simulation

```
> set.seed(0)    # fix seed value to get the same sample every time.
> n.sample = rnorm(n = 10000, mean = 55, sd = 4.5)
> install.packages("moments")       # install package "moments".
> library(moments)                  # load package "moments".
> skewness(n.sample)                # calculate skewness.
```

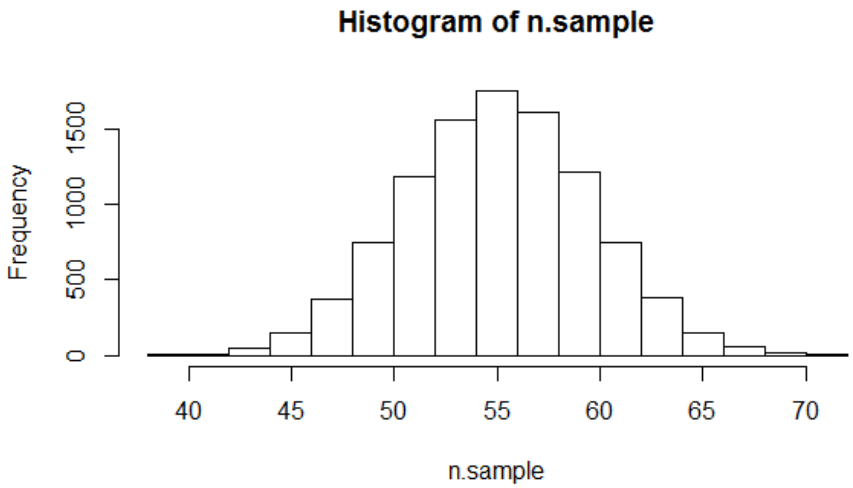
[1] -0.00236372

```
> kurtosis(n.sample)                # calculate kurtosis.
```

[1] 2.907761

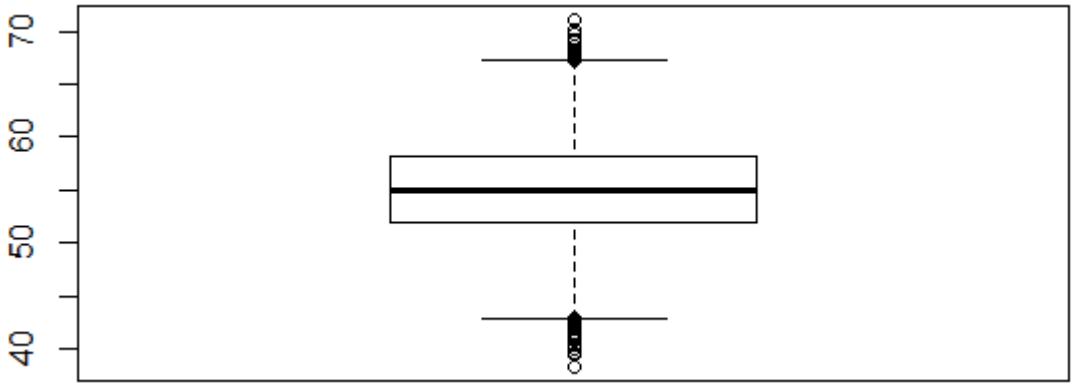
Draw Histogram using hist()function.

```
> hist(n.sample)
```



Produce boxplot using boxplot()function.

> boxplot(n.sample)



Linear Model

Suppose we want to model the response Y in terms of three predictors, X_1, X_2 and X_3. One general form for the model would be:

$$Y = f(X_1, X_2, X_3) + \varepsilon$$

where f is some unknown function and ε is the error. ε is additive in this instance, but could enter in some even more general form. If we assume that f is a smooth, continuous function that still leaves a very wide range of possibilities. Even with just three predictors, we typically may not have enough data to try to estimate f directly. So we usually have to assume that it has some more restricted form, perhaps linear as in:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \varepsilon$$

where β_i, i= 0, 1, 2, 3 are unknown parameters. β_0 is called the intercept term. Thus the problem is reduced to the estimation of four parameters rather than the infinite dimensional f. In a linear model the parameters enter linearly - the predictors themselves do not have to be linear. For example:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 \log X2 + \beta_3 X_1 X_3 + \varepsilon$$

is a linear model, but

$$Y = \beta_0 + \beta_1 X_1^{\beta 2} + \varepsilon$$

is not.

Linear Regression Model

Regression analysis is a statistical methodology that utilizes the relation between two or more quantitative variables so that one variable can be predicted from the other, or others. This methodology is widely used in business, the social and behavioral sciences, the biological sciences including agriculture and fishery research. Depending on the nature of the relationships between X and Y, regression approach may be classified into two broad categories viz., linear regression models and nonlinear regression models. The response variable is generally related to other causal variables through some parameters. The models that are linear in these parameters, as discussed in the previous section, are known as linear models, whereas in nonlinear models parameters are appear nonlinearly.

The generic form of a linear regression model is $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p + \varepsilon$.

Example

An experiment was conducted to study the hybrid seed production of bottle gourd (*Lagenaria siceraria (Mol.) Standl.*) Cv. Pusa hybrid-3 under open field conditions during Kharif-2005 at ICAR-Indian Agricultural Research Institute, New Delhi. The main aim of the investigation was to compare natural pollination and hand pollination under field conditions. The data were collected on 10 randomly selected plants from each of natural pollination and hand pollination. The data were collected on number of fruit set for the period of 45 days, fruit weight (kg), seed yield per plant (g) and seedling length (cm). The data obtained is as given below:

{Here 1 denotes natural pollination and 2 denotes the hand pollination)

Group	No of Fruit set (45 days)	Fruit Weight(Kg)	Seed Yield per plant(g)	Seedling length(cm)
1	7	1.85	147.70	16.86
1	7	1.86	136.86	16.77
1	6	1.83	149.97	16.35
1	7	1.89	172.33	18.26
1	7	1.80	144.46	17.90

1	6	1.88	138.3	16.95
1	7	1.89	150.58	18.15
1	7	1.79	140.99	18.86
1	6	1.85	140.57	18.39
1	7	1.84	138.33	18.58
2	6.3	2.58	224.26	18.18
2	6.7	2.74	197.50	18.07
2	7.3	2.58	230.34	19.07
2	8	2.62	217.05	19
2	8	2.68	233.84	18
2	8	2.56	216.52	18.49
2	7.7	2.34	211.93	17.45
2	7.7	2.67	210.37	18.97
2	7	2.45	199.87	19.31
2	7.3	2.44	214.30	19.36

The function used for regression analysis in R is $lm(y \sim x_1+x_2+x_3+\ldots+x_p)$. To fit a multiple linear regression by taking seed yield per plant (sy) as dependent variable and number of fruit set (fs), fruit weight (fw) and seedling length (sl) as explanatory variables, use following commands:

```
> op=lm(formula = sy ~ fs + fw + sl)

> op

Call:

lm(formula = sy ~ fs + fw + sl)

Coefficients:

(Intercept)          fs          fw          sl

  -71.2001      7.2949     85.2960      0.6792
```

summary() gives other information related to the regression analysis.

```
> summary(op)

Call:

lm(formula = sy ~ fs + fw + sl)

Residuals:

    Min      1Q  Median      3Q     Max
```

-26.160 -6.226 -1.820 10.397 18.854

Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-71.2001	65.0731	-1.094	0.290
fs	7.2949	5.7217	1.275	0.221
fw	85.2960	9.9705	8.555	2.3e-07 ***
sl	0.6792	3.9812	0.171	0.867

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 13.02 on 16 degrees of freedom

Multiple R-squared: 0.8975, Adjusted R-squared: 0.8782

F-statistic: 46.68 on 3 and 16 DF, p-value: 3.887e-08

From the output we see that the intercept is estimated to -71.2001 with a standard error of 65.0731.

Confidence intervals for the regression parameters may be computed by the function confint. As default confint computes 95% confidence intervals.

> confint(op)

The predicted values and residuals are extracted from linear regression as follows:

	2.5 %	97.5 %
(Intercept)	-209.148953	66.748804
fs	-4.834573	19.424435
fw	64.159471	106.432548
sl	-7.760656	9.119007

The predicted values and residuals are extracted from linear regression as follows:

> pred = predict(op)

> pred

```
       1        2        3        4        5        6        7        8        9
149.1130 149.9048 139.7657 153.4756 145.5545 144.4380 153.4009 145.3536 142.8572
      10       11       12       13       14       15       16       17       18
149.4282 207.1691 223.6597 215.0685 223.5393 227.9778 218.0751 196.4152 225.5952
      19       20
201.9545 203.3240
> res = resid(op)
> res
         1          2          3          4          5          6          7
 -1.412960 -13.044795  10.204270  18.854354  -1.094502  -6.138035  -2.820937
         8          9         10         11         12         13         14
 -4.363551  -2.287168 -11.098182  17.090893 -26.159732  15.271495  -6.489255
        15         16         17         18         19         20
  5.862160  -1.555115  15.514829 -15.225200  -2.084546  10.975976
```

References

Faraway, J. J. (2016). *Linear models with R*. Chapman and Hall/CRC.

Goon, A. M., Gupta, M. K. and Dasgupta, R. (1986). *Outline of Statistics.* Vol. I. World Press.

Goon, A. M., Gupta, M. K. and Dasgupta, R. (2008). *Fundamentals of Statistics.* Vol. I. Atlantic Publishers.

https://en.wikipedia.org/wiki/Central_tendency

https://socialresearchmethods.net/kb/statdesc.php

https://statistics.laerd.com/statistical-guides/measures-central-tendency-mean-mode-median.php

https://statisticsbyjim.com/basics/measures-central-tendency-mean-median-mode/

https://towardsdatascience.com/understanding-descriptive-statistics-c9c2b0641291

https://www.toppr.com/guides/business-mathematics-and-statistics/measures-of-central-tendency-and-dispersion/measure-of-dispersion/

Transcriptome Data Analysis

De novo Assembly

Genome assembly refers to the process of taking a large number of short DNA sequences and putting them back together to create a representation of the original chromosomes from which the DNA originated. De novo genome assemblies assume no prior knowledge of the source DNA sequence length, layout or composition. In a genome sequencing project, the DNA of the target organism is broken up into millions of small pieces and read on a sequencing machine. These "reads" vary from 20 to 1000 nucleotide base pairs (bp) in length depending on the sequencing method used. Typically for Illumina type short read sequencing, reads of length 36 - 150 bp are produced. These reads can be either "single ended" as described above or "paired end." Paired end reads are produced when the fragment size used in the sequencing process is much longer (typically 250 - 500 bp long) and the ends of the fragment are read in towards the middle. This produces two "paired" reads.

The goal of a sequence assembler is to produce long contiguous pieces of sequence (contigs) from these reads. The contigs are sometimes then ordered and oriented in relation to one another to form scaffolds. The distances between pairs of a set of paired end reads is useful information for this purpose. The mechanisms used by assembly software are varied but the most common type for short reads is assembly by de Bruijn graph

Determining the DNA sequence of an organism is useful in fundamental research into why and how they live, as well as in applied subjects. Because of the importance of DNA to living things, knowledge of a DNA sequence may be useful in practically any biological research. For example, in medicine it can be used to identify, diagnose and potentially develop treatments for genetic diseases. Similarly, research into pathogens may lead to treatments for contagious diseases

The protocol in a nutshell:

- Obtain sequence read file (s) from sequencing machine (s).
- Look at the reads - get an understanding of what you've got and what the quality is like.
- Raw data cleanup/quality trimming if necessary.
- Choose an appropriate assembly parameter set.
- Assemble the data into contigs/scaffolds.
- Examine the output of the assembly and assess assembly quality.

Raw read sequence file formats.

Raw read sequences can be stored in a variety of formats. The reads can be stored as text in a Fasta file or with their qualities as a FastQ file. They can also be stored as alignments to references in other formats such as SAM or its binary compressed implementation BAM. The entire file formats (with the exception of the binary BAM format) can be compressed easily and often are stored so (.gz for gzipped files.)

The most common read file format is FastQ as this is what is produced by the Illumina sequencing pipeline. This will be the focus of our discussion henceforth.

Section 1: Read Quality Control

Examine the quality of your raw read files.

We check the following parameters from fastq file

- Base quality score distribution
- Sequence quality score distribution
- Average base content per read
- GC distribution in the reads
- Check for over-represented sequences

Quality trimming/cleanup of read files.

Based on quality of sequence reads, we trimmed sequence reads where necessary, to retain only high quality sequence for further analysis. In addition, the low-quality sequence reads were excluded from the analysis from the Trimmed single-end reads; we removed unwanted sequences, adapter sequences and others. The trimming and contamination removal step is done by Trimmomatic software.

Trimmomatic should produce 2 pair files (1 left and 1 right hand end) and 1 or 2 single "orphaned reads" files if you trimmed a pair of read files using paired end mode. It only produces 1 output read file if you used it in single ended mode. Each read library (2 paired files or 1 single ended file) should be trimmed separately with parameters dependent on their own FastQC reports. The output files are the ones you should use for assembly.

Section 2: Assembly

The purpose of this section of the protocol is to outline the process of assembling the quality trimmed reads into draft contigs. Most assembly software has a number of input parameters which need to be set prior to running. These parameters can and do have a large effect on the

outcome of any assembly. Assemblies can be produced which have less gaps, less or no mis-assemblies, less errors by tweaking the input parameters. Therefore, knowledge of the parameters and their effects is essential to getting good assemblies. In most cases an optimum set of parameters for your data can be found using an iterative method.

To generate a reference assembly that we can later use for analysing differential expression, we'll combine the read data sets for the different conditions together into a single target for Trinity assembly. We do this by providing Trinity with a list of all the left and right fastq files to the --left and --right parameters as comma-delimited like so:

```
cat paired_sample1_R1.fastq paired_sample2_R2.fastq>p_sample1_sample2_R1.fastq

cat paired_sample1_R2.fastq paired_sample2_R2.fastq>p_sample1_sample2_R2.fastq

Trinity --seqType fq --max_memory 50G --left p_sample1_sample2_R1.fastq --right
p_sample1_sample2_R2.fastq --CPU 6 --output Trinity.fasta
```

Running Trinity on this data set may take 10 to 15 minutes. You'll see it progress through the various stages, starting with Jellyfish to generate the k-mer catalog, then followed by Inchworm to assemble 'draft' contigs, Chrysalis to cluster the contigs and build de Bruijn graphs, and finally Butterfly for tracing paths through the graphs and reconstructing the final isoform sequences.

Just to look at the top few lines of the assembled transcript fasta file, you can run:

```
% head trinity_out_dir/Trinity.fasta
```

and you can see the Fasta-formatted Trinity output:

```
>TRINITY_DN506_c0_g1_i1 len=171 path=[149:0-170] [-1, 149, -2]
TGAGTATGGTTTTGCCGGTTTGGCTGTTGGTGCAGCTTTGAAGGGCCTAAAGCCA
ATTGT
TGAATTCATGTCATTCAACTTCTCCATGCAAGCCATTGACCATGTCGTTAACTCG
GCAGC
AAAGACACATTATATGTCTGGTGGTACCCAAAAATGTCAAATCGTGTTCAG
>TRINITY_DN512_c0_g1_i1 len=168 path=[291:0-167] [-1, 291, -2]
ATATCAGCATTAGACAAAAGATTGTAAAGGATGGCATTAGGTGGTCGAAGTTTC
AGGTCT
AAGAAACAGCAACTAGCATATGACAGGAGTTTTGCAGGCCGGTATCAGAAATTG
CTGAGT
AAGAACCCATTCATATTCTTTGGACTCCCGTTTTGTGGAATGGTGGTG
>TRINITY_DN538_c0_g1_i1 len=310 path=[575:0-309] [-1, 575, -2]
GTTTTCCTCTGCGATCAAATCGTCAAACCTTAGACCTAGCTTGCGGTAACCAGAG
TACTT
```

The FASTA sequence header for each of the transcripts contains the identifier for the transcript (eg. 'TRINITY_DN506_c0_g1_i1'), the length of the transcript, and then some information about how the path was reconstructed by the software by traversing nodes within the graph.

It is often the case that multiple isoforms will be reconstructed for the same 'gene'. Here, the 'gene' identifier corresponds to the prefix of the transcript identifier, such as 'TRINITY_DN506_c0_g1', and the different isoforms for that 'gene' will contain different isoform numbers in the suffix of the identifier (eg. TRINITY_DN506_c0_g1_i1 and TRINITY_DN506_c0_g1_i2 would be two different isoform sequences reconstructed for the single gene TRINITY_DN506_c0_g1). It is useful to perform certain downstream analyses, such as differential expression, at both the 'gene' and at the 'isoform' level, as we'll do later below.

Evaluating the assembly

There are several ways to quantitatively as well as qualitatively assess the overall quality of the assembly.

Examine assembly stats

Capture some basic statistics about the Trinity assembly:

```
% $TRINITY_HOME/util/TrinityStats.pl trinity_out_dir/Trinity.fasta
```

Which should generate data like so. Note your numbers may vary slightly, as the assembly results are not deterministic.

```
################################
## Counts of transcripts, etc.
################################
Total trinity 'genes':          683
Total trinity transcripts:      687
Percent GC: 44.39

########################################
Stats based on ALL transcript contigs:
########################################

Contig N10: 742
Contig N20: 525
Contig N30: 423
Contig N40: 346
Contig N50: 300
```

```
Median contig length: 216
Average contig: 279.85
Total assembled bases: 192257

#####################################################
## Stats based on ONLY LONGEST ISOFORM per 'GENE':
#####################################################

Contig N10: 728
Contig N20: 524
Contig N30: 420
Contig N40: 343
Contig N50: 296

Median contig length: 215
Average contig: 278.14
Total assembled bases: 189969
```

The 'N50 statistic indicates that at least half of the assembled bases are in contigs of at least that contig length'. We extend the N50 statistic to provide N40, N30, etc. statistics with similar meaning. As the N-value is decreased, the corresponding length will increase.

Transcript expression suing RSEM

To estimate the expression levels of the Trinity-reconstructed transcripts, we use the strategy supported by the RSEM software involving read alignment followed by expectation maximization to assign reads according to maximum likelihood estimates. In essence, we first align the original rna-seq reads back against the Trinity transcripts, then run RSEM to estimate the number of rna-seq fragments that map to each contig. Because the abundance of individual transcripts may significantly differ between samples, the reads from each sample (and each biological replicate) must be examined separately, obtaining sample-specific abundance values.

We include a script to faciliate running of RSEM on Trinity transcript assemblies. The script we execute below will run the Bowtie aligner to align reads to the Trinity transcripts, and RSEM will then evaluate those alignments to estimate expression values. Again, we need to run this separately for each sample and biological replicate (ie. each pair of fastq files).

Let's start with one of the GSNO treatment fastq pairs like so:

```
%$TRINITY_HOME/util/align_and_estimate_abundance.pl –transcripts Trinity.fasta --seqType fq --left paired_sample1_R1.fastq --right paired_sample1_R2.fastq --est_method RSEM --aln_method bowtie2 --trinity_mode --prep_reference --output_dir rsem_outdir_sample1 &
```

```
%$TRINITY_HOME/util/align_and_estimate_abundance.pl –transcripts Trinity.fasta --seqType fq --left paired_sample2_R1.fastq --right paired_sample2_R2.fastq --est_method RSEM --aln_method bowtie2 --trinity_mode --prep_reference --output_dir rsem_outdir_sample2 &
```

The primary output generated by RSEM is the file containing the expression values for each of the transcripts. Examine this file like so:

transcript_id	gene_id	length	effective_length	expected_count	TPM	FPKM	IsoPct
TRINITY_DN0_c0_g1_i1	TRINITY_DN0_c0_g1	328	198.75	29.00	9093.16	43883.19	100.00
TRINITY_DN0_c0_g2_i1	TRINITY_DN0_c0_g2	329	199.75	0.00	0.10	0.48	100.00
TRINITY_DN100_c0_g1_i1	TRINITY_DN100_c0_g1	198	69.79	1.00	892.99	4309.53	100.00
TRINITY_DN101_c0_g1_i1	TRINITY_DN101_c0_g1	233	104.12	2.00	1197.08	5777.04	100.00
TRINITY_DN102_c0_g1_i1	TRINITY_DN102_c0_g1	198	69.79	0.00	0.00	0.00	0.00
TRINITY_DN103_c0_g1_i1	TRINITY_DN103_c0_g1	346	216.72	7.00	2012.95	9714.40	100.00
TRINITY_DN104_c0_g1_i1	TRINITY_DN104_c0_g1	264	134.91	1.00	461.94	2229.29	100.00
TRINITY_DN105_c0_g1_i1	TRINITY_DN105_c0_g1	540	410.62	19.00	2883.65	13916.35	100.00
TRINITY_DN106_c0_g1_i1	TRINITY_DN106_c0_g1	375	245.67	3.00	761.01	3672.58	100.00

The key columns in the above RSEM output are the transcript identifier, the 'expected_count' corresponding to the number of RNA-Seq fragments predicted to be derived from that transcript, and the 'TPM' or 'FPKM' columns, which provide normalized expression values for the expression of that transcript in the sample.

The FPKM expression measurement normalizes read counts according to the length of transcripts from which they are derived (as longer transcripts generate more reads at the same expression level), and normalized according to sequencing depth. The FPKM acronym stand for 'fragments per kilobase of cDNA per million fragments mapped'.

TPM 'transcripts per million' is generally the favored metric, as all TPM values should sum to 1 million, and TPM nicely reflects the relative molar concentration of that transcript in the sample. FPKM values, on the other hand, do not always sum to the same value, and do not have the similar property of inherrently representing a proportion within a sample, making

comparisons between samples less straightforward. TPM values can be easily computed from FPKM values like so: TPMi = FPKMi / (sum all FPKM values) * 1 million.

Running this on all the samples can be montonous, and with many more samples, advanced users would generally write a short script to fully automate this process. We won't be scripting here, but instead just directly execute abundance estimation just as we did above but on the other five pairs of fastq files. We'll examine the outputs of each in turn, as a sanity check and also just for fun

Generate a transcript counts matrix and perform cross-sample normalization:

Now, given the expression estimates for each of the transcripts in each of the samples, we're going to pull together all values into matrices containing transcript IDs in the rows, and sample names in the columns. We'll make two matrices, one containing the estimated counts, and another containing the TPM expression values that are cross-sample normalized using the TMM method. This is all done for you by the following script in Trinity, indicating the method we used for expression estimation and providing the list of individual sample abundance estimate files:

```
%$TRINITY_HOME/util/abundance_estimates_to_matrix.pl sample1.isoforms.results sample2.isoforms.results --est_method RSEM  --out_prefix sample1_sample2.
```

You should find a matrix file called 'Trinity_trans.counts.matrix', which contains the counts of RNA-Seq fragments mapped to each transcript. Examine the first few lines of the counts matrix:

The counts matrix will be used by edgeR (or other tools in Bioconductor) for statistical analysis and identifying significantly differentially expressed transcripts.

Differential expression using edgeR

The edgeR software is part of the R Bioconductor package, and we provide support for using it in the Trinity package.

Having biological replicates for each of your samples is crucial for accurate detection of differentially expressed transcripts. In our data set, we have three biological replicates for each of our conditions, and in general, having three or more replicates for each experimental condition is highly recommended.

To detect differentially expressed transcripts, run the Bioconductor package edgeR using our counts matrix:

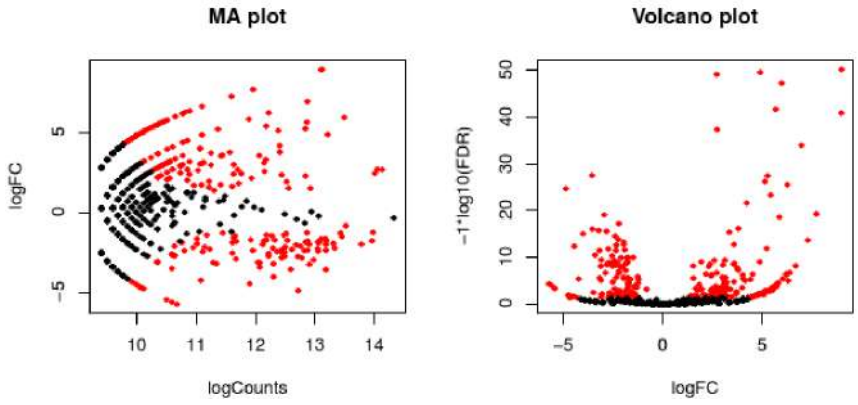
```
%$TRINITY_HOME/Analysis/DifferentialExpression/run_DE_analysis.pl --matrix sample1_sample2.counts.matrix --method edgeR --output sample1_sample2_DEG --dispersion 0.05
```

The files '*.DE_results' contain the output from running EdgeR to identify differentially expressed transcripts in each of the pairwise sample comparisons. Examine the format of one of the files, such as the results from comparing Sp_log to Sp_plat:

	logFC	logCPM	PValue	FDR
TRINITY_DN530_c0_g1_i1	8.98959066402695	13.1228060448362	8.32165055033775e-54	5.54221926652494e-51
TRINITY_DN589_c0_g1_i1	4.89839016049723	13.2154341051504	7.57411107973887e-53	2.52217898955304e-50
TRINITY_DN44_c0_g1_i1	2.69777851490106	14.1332252828559	5.23937214153746e-52	1.16314061542132e-49
TRINITY_DN219_c0_g1_i1	5.96230500956404	13.4919132973162	5.11512415417842e-51	8.51668171670708e-49
TRINITY_DN513_c0_g1_i1	5.67480055313841	12.8660937412604	1.54064866426519e-44	2.05214402080123e-42
TRINITY_DN494_c0_g1_i1	8.97722926993194	13.108274725098	3.6100707178792e-44	4.00717849684591e-42
TRINITY_DN365_c0_g1_i1	2.71537635410452	14.0482419858984	8.00431159168039e-41	7.61553074294163e-39
TRINITY_DN415_c0_g1_i1	6.96733684710045	12.875060733337	3.67004658844109e-36	3.05531378487721e-34
TRINITY_DN59_c0_g1_i1	-3.57509574692798	13.1852604213653	3.74452542871713e-30	2.77094881725068e-28

These data include the log fold change (logFC), log counts per million (logCPM), P- value from an exact test, and false discovery rate (FDR).

The EdgeR analysis above generated both MA and Volcano plots based on these data.



The red data points correspond to all those features that were identified as being significant with an FDR <= 0.05.

Trinity facilitates analysis of these data, including scripts for extracting transcripts that are above some statistical significance (FDR threshold) and fold-change in expression, and generating figures such as heatmaps and other useful plots, as described below.

Reference Based Assembly

Reference based assembly is performed when reference genome is available for differential gene expression analysis.

Outline

1. Quality filter datasets using **Trimmomatic**.
2. Assess data quality using **FastQC**.
3. Align the RNA-seq reads to the human genome using **TopHat2**.
4. Assemble transcripts based on RNA-seq data using **cufflinks** and **cuffmerge**.
5. Compare expression differences using **cuffdiff**.

Quality Control, Alignment, and Differential Gene Expression

There are 4 RNA-seq datasets that we'll use in this exercise. Examine a few lines of each library using a different command, such as`more`, `less`, `head`, and `tail`.

- `brain_1_fastq.txt`
- `brain_2_fastq.txt`
- `adrenal_1_fastq.txt`
- `adrenal_2_fastq.txt`

2. **Assess the quality of the data before quality filtering using FastQC:**

3. **Trim adapter sequences and quality filter the RNA-seq data (fastq files) using Trimmomatic:**

Trim adapter sequences and quality filter each of the 4 datasets using Trimmomatic.

```
$ Trimmomatic-0.32/trimmomatic-0.32.jar PE -phred33 /brain1_1.fastq brain1_2.fastq
P_brain1_1.fastq U_brain1_1.fastq P_brain1_2.fastq U_brain1_2.fastq
ILLUMINACLIP:/opt/software/Trimmomatic-0.32/adapters/TruSeq3-PE.fa:2:30:10
LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15 MINLEN:36
```

4. **Examine each of the files after adapter trimming and quality filtering using the** `more` **or** `less` **commands.**

5. **Assess the quality of the data after Trimmomatic quality filtering using FastQC:**

6. **Create a** bowtie index **for the human genome** (NOTE: THIS STEP WAS DONE IN ADVANCE OF CLASS, BUT YOU WILL NEED TO CREATE A C ELEGANS BOWTIE INDEX FOR THE ASSIGNMENT)**:**

```
$ bowtie2-build genome.fa human
```

7. **Run TopHat to align sequences from each of the libraries to the human genome** (you will run tophat 4 times in total on the 4 fastq files listed below)**:**

```
$ tophat -G 'path_to_genome_annotations.gtf' -o 'output_folder''
path_to_bowtie_index_for_reference_genome/prefix' 'fastq_file'
```

- The directory containing the fastq files is the current working directory: /Users/graduatestudent/Documents/RNA-seq.
- Name the output folders as follows: adrenal1; adrenal2; brain1; brain2
- The path to the bowtie index is /Users/graduatestudent/Documents/RNA-seq/hg19_bowtie2/ and the prefix is human.
- The path to the genome annotations file is /Users/graduatestudent/Documents/RNA-seq/hg19_bowtie2/hg19_chr19_gene_annotation.gtf.

8. **Determine what proportion of the reads from each library were aligned:**

Use the UNIX more command to open each TopHat summary file in the terminal. The TopHat summary files are named align_summary.txt and are located in the output folder specified in step 5.

```
$ more ./adrenal1/align_summary.txt

$ more ./adrenal2/align_summary.txt

$ more ./brain1/align_summary.txt

$ more ./brain2/align_summary.txt
```

9. **Run cufflinks to assemble transcripts** (cufflinks uses the accepted_hits.bam output files from TopHat)**:**

```
$ cufflinks -o 'output_folder1' 'path_to_library1_accepted_hits.bam'
$ cufflinks -o 'output_folder2' 'path_to_library2_accepted_hits.bam'
```

```
$ cufflinks -o 'output_folder3' 'path_to_library3_accepted_hits.bam'
$ cufflinks -o 'output_folder4' 'path_to_library4_accepted_hits.bam'
```

Name the output folders as follows:
output_folder1: cufflinks_adrenal1
output_folder2: cufflinks_adrenal2
output_folder3: cufflinks_brain1
output_folder4: cufflinks_brain2.

The paths to the accepted_hits.bam files from the RNA-seq_Data directory are as follows:path_to_library1_accepted_hits.bam: ./adrenal1/accepted_hits.bam
path_to_library2_accepted_hits.bam: ./adrenal2/accepted_hits.bam
path_to_library3_accepted_hits.bam: ./brain1/accepted_hits.bam
path_to_library4_accepted_hits.bam: ./brain2/accepted_hits.bam

10. **Merge the assembled transcripts from the four libraries using cuffmerge:**

Create a file called assemblies.txt with the paths to each of the individual assemblies files:

```
$ echo ./cufflinks_adrenal1/transcripts.gtf >assemblies.txt
$ echo ./cufflinks_adrenal2/transcripts.gtf >>assemblies.txt
$ echo ./cufflinks_brain1/transcripts.gtf >>assemblies.txt
$ echo ./cufflinks_brain2/transcripts.gtf >>assemblies.txt
```

Merge assemblies:

```
$ cuffmerge -g 'path_to_genome_annotations.gtf' -s 'path_to_genome_sequence.fa'
assemblies.txt
```

- The path to the genome annotations file is /Users/graduatestudent/Documents/RNA-seq/hg19_bowtie2/hg19_chr19_gene_annotation.gtf
- The path to the human genome sequence is /Users/graduatestudent/Documents/RNA-seq/hg19_bowtie2/human.fa

You should now have a single file, merged.gtf located in a folder called merged_asm that was created by cuffmerge, that contains all of the predicted transcripts based on the sequencing data.

11. **Run cuffdiff to identify genes differentially regulated between the adrenal and brain tissue samples:**

- Provide an output_folder name such as cuffdiff_output
- The path_to_merged.gtf is merged_asm/merged.gtf
- You will compare the two adrenal libraries to the two brain libraries. Cufflinks uses the accepted_hits.bam output files from TopHat. If you are in the RNA-seq_Data directory, the paths to these files are as follows:
  - ./adrenal1/accepted_hits.bam
  - ./adrenal2/accepted_hits.bam
  - ./brain1/accepted_hits.bam
  - ./brain2/accepted_hits.bam

EXAMPLE

```
$ cuffdiff -o cuffdiff_output -L adrenal,brain merged_asm/merged.gtf
./adrenal1/accepted_hits.bam,./adrenal2/accepted_hits.bam
\./brain1/accepted_hits.bam,./brain2/accepted_hits.bam

$ cuffdiff -o 'output_folder' -L adrenal, brain 'path_to_merged.gtf'
'library1_replicate1'/accepted_hits.bam,./'library1_replicate2'/accepted_hits.bam
\'library2_replicate1'/accepted_hits.bam,./'library2_replicate2'/accepted_hits.bam
```

Several output files are generated. Explore these on your own. The gene_exp.diff file contains a summary of differential gene expression.

12. **Identify which genes are differentially expressed in adrenal and brain tissue:**

- Open the gene_exp.diff file from step 9 using Excel.
- Sort the data in Excel based on the q value.

R-Graphics

(Practical)

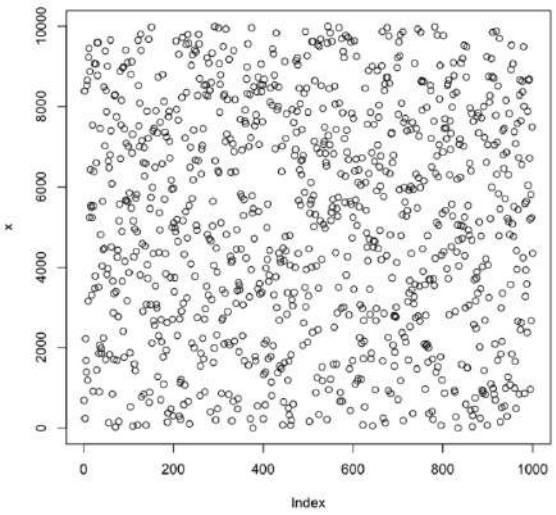
A picture says thousand words. When it comes down to the visualization of the data, R has an edge over the existing programming languages. This chapter provides the most basic information to get started producing plots in R.

plot()is the main graphing function. It automatically produces simple plots for vectors, functions or data frames.

Plotting a Vector

plot(x) will print the elements of the vector x according to their index.

```
> x<-sample(1:10000,1000,replace=F)
> plot(x)
```




```
> plot (sort(x))
```

Common parameters for plot ()

- *Specifying labels:*

  •main – provides a title

  •xlab – label for the x axis

  •ylab – label for the y axis

- *Specifying range limits:*

•ylim – 2-element vector gives range for x axis

•`xlim` – 2-element vector gives range for y axis

- It can be seen that the first plot is of circular points and black in colour. This is the default colour.
- We can change the plot type with the argument `type`. It accepts the following strings and has the given effect.

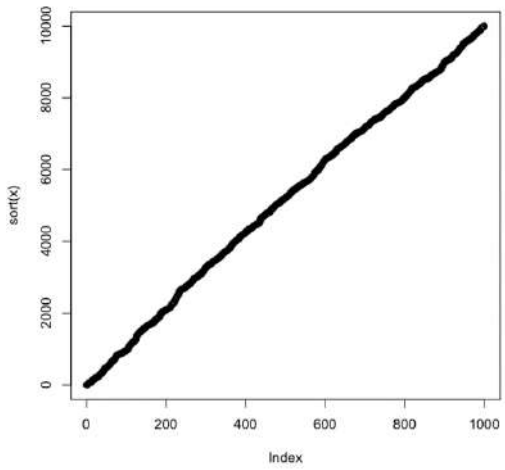
"p" - points

"l" - lines

"b" - both points and lines

"c" - empty points joined by lines

"o" - overplotted points

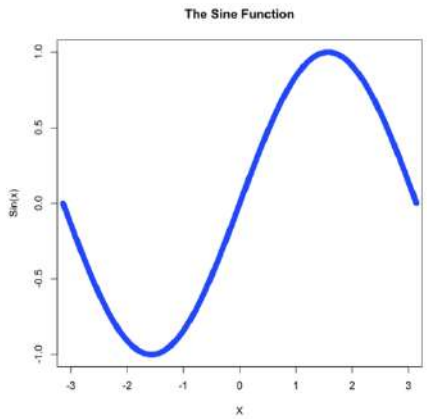


"h" - histogram like vertical lines

"n" - does not produce any points or lines

"s" and "S" – stair steps

```
> x<- seq(-pi,pi,.001)
> plot(x,sin(x),main= "The Sine Function",xlab="X",
+ ylab="Sin(x)",type= "b",col= "blue")
```

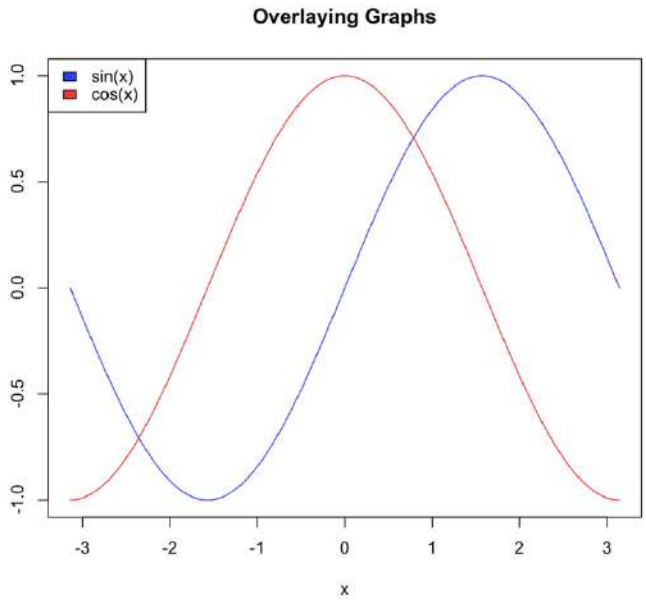


Overlaying Plots Using legend() **function**

Calling plot() multiple times will have the effect of plotting the current graph on the same window replacing the previous one. However, sometimes it is required to overlay the plots in order to compare the results.

This is made possible with the functions lines() and points() to add lines and points respectively, to the existing plot.

```
> plot(x, sin(x),main="Overlaying Graphs",ylab="",type="l",col="blue")
> lines(x,cos(x),col="red")
> legend("topleft",c("sin(x)","cos(x)"),fill=c("blue","red"))
```

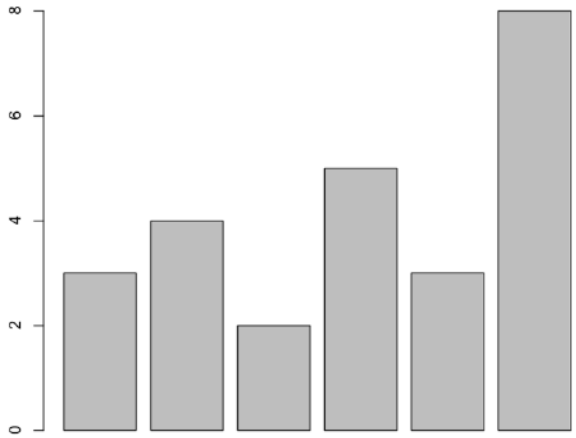


What pch argument does in plot function?

Bar Plot

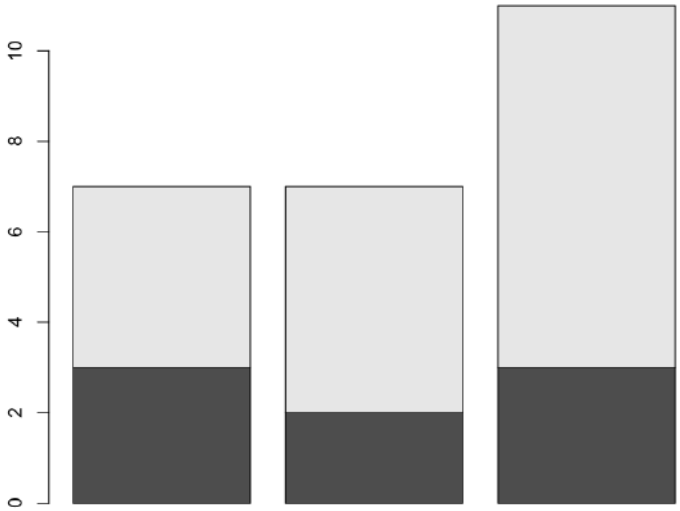
- Barplots can be created with the barplot(height) function, where *height* is a vector or matrix.
- If it is a vector, the values determine the heights of the bars in the plot.

```
> barplot(c(3,4,2,5,3,8))
```



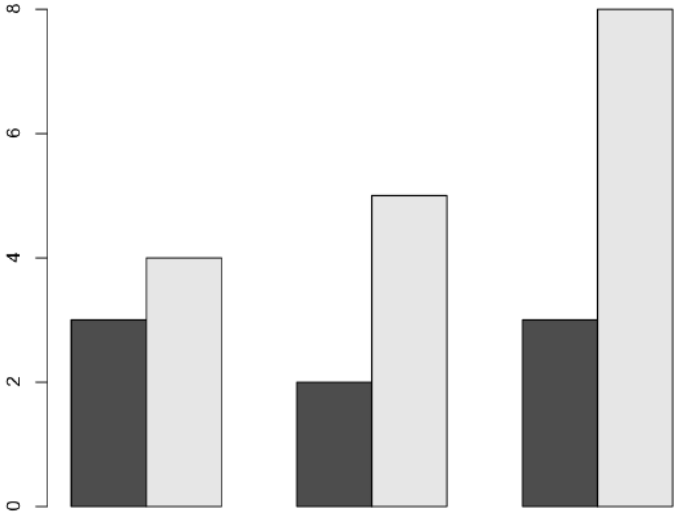
- If height is a matrix and the option, `beside=FALSE` then each bar of the plot corresponds to a column of height, with the values in the column giving the heights of stacked "sub-bars".

```
> barplot(matrix(c(3,4,2,5,3,8),nrow=2))
```

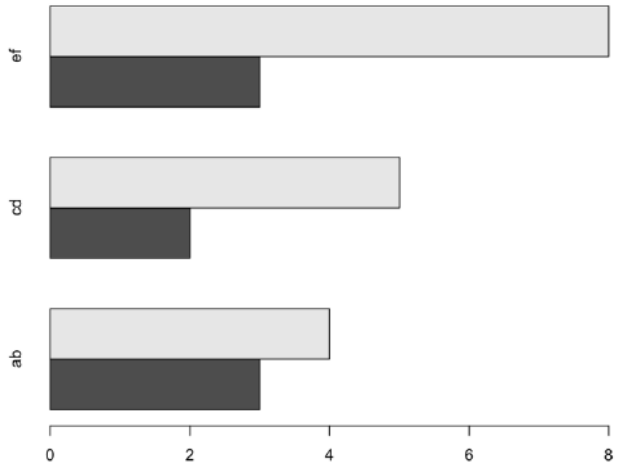


- If height is a matrix and beside=TRUE, then the values in each column are juxtaposed rather than stacked. By including the option names.arg=(character vector) to label the bars. The option horiz=TRUE to createa a horizontal barplot.

```
> barplot(matrix(c(3,4,2,5,3,8),nrow=2),beside=T)
```




```
> barplot(matrix(c(3,4,2,5,3,8),nrow=2),beside=T,names.arg= c("ab","cd","ef"))

> barplot(matrix(c(3,4,2,5,3,8),nrow=2),beside=T,names.arg= c("ab","cd","ef"),horiz=T)
```

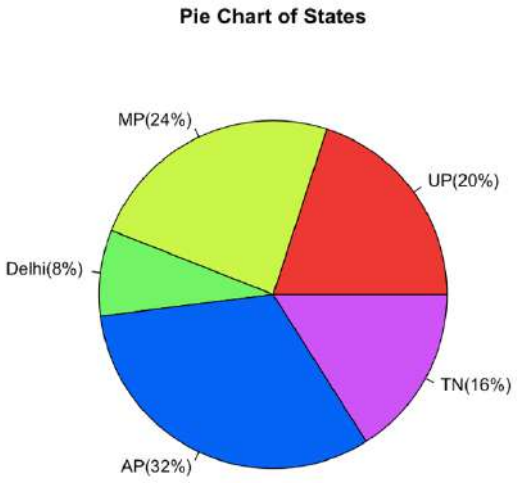


Can you change the colours in bar-plot?

Pie Chart

Pie-chart is usually not recommended. However, it can be used in case of percentage and/or proportion data.

```
> slices <- c(10, 12, 4, 16, 8)
> lbls <- c("UP", "MP", "Delhi", "AP", "TN")
> pct <- round(slices/sum(slices)*100)
> lbls<- paste(lbls,"(",pct,"%",")",sep="")
> pie(slices,labels = lbls,col=rainbow(length(lbls)),main="Pie Chart of States")
```

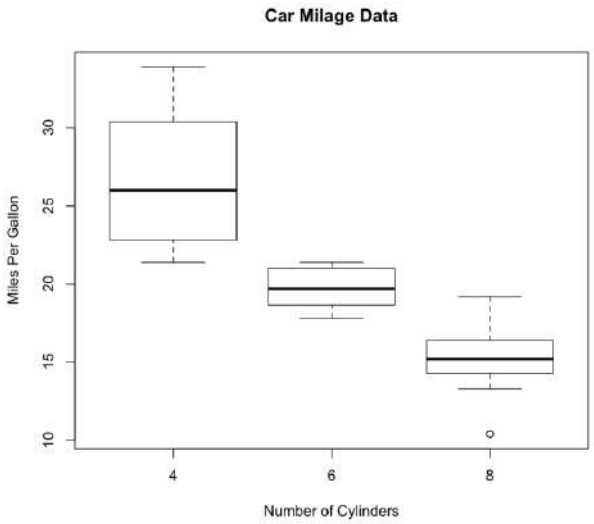


Are you interested in 3-D plot? Try `pie3D` function in `plotrix` package.

Box Plot

A boxplot is a standardized way of displaying the distribution of data based on a five number summary ("minimum", first quartile (Q1), median, third quartile (Q3), and "maximum"). It can tell about the outliers and what their values are. It can also tell if the data is symmetrical.

```
> boxplot(mpg~cyl,data=mtcars, main="Car Milage Data",
+ xlab="Number of Cylinders", ylab="Miles Per Gallon")
```

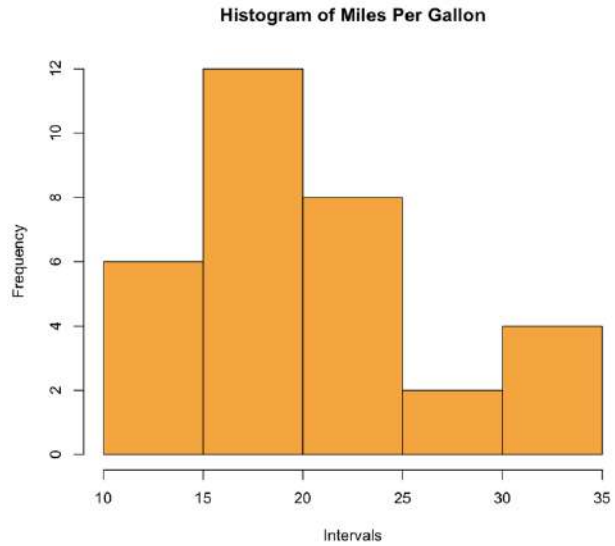


- Can you tell why there are 3 boxes only?
- Is there any outlier present in the plot?
- Can you spot the second quartile?

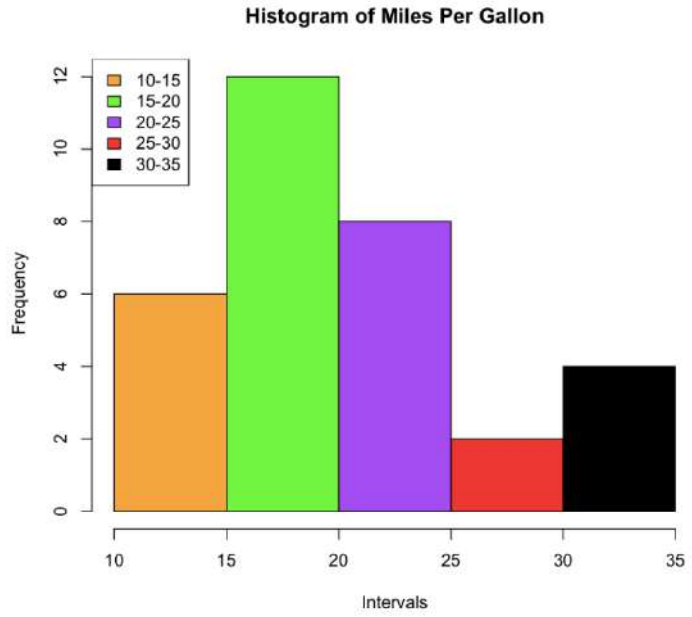
Histogram and Density Plot

Histogram is a very important tool to know about the distribution of the data.

```
> hist(mtcars$mpg,main= "Histogram of Miles Per Gallon",
+ xlab="Intervals",breaks=5, col="orange")
```

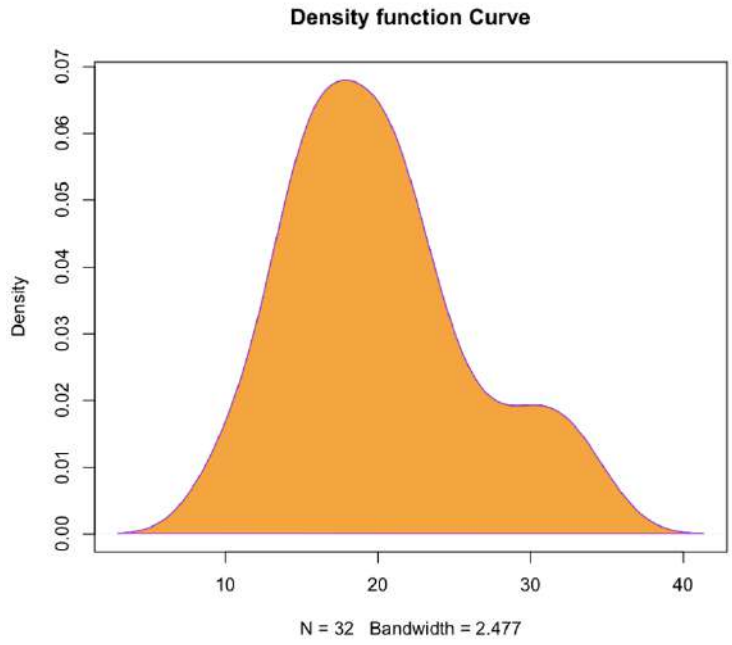


```
> hist(mtcars$mpg,main= "Histogram of Miles Per Gallon",
+ xlab="Intervals",breaks=5, col=c("orange","green","purple","red","black"))
> legend("topleft",c("10-15","15-20","20-25","25-30","30-35"),
+ fill=c("orange","green","purple","red","black"))
```



Density plot can be made by the help of `density` function.

```
> var<-mtcars$mpg
> den<-density(var)
> plot(den, main= 'Density function Curve')
> polygon(den, col= "orange", border= "purple")
```



Test of Significance

1. Introduction

In applied investigations, one is often interested in comparing some characteristic (such as mean or variance) of a group with a specified value, or in comparing two or more groups with regard to the characteristic. For instance, one may want to know whether mean timber yield obtained from recently felled plantations of a particular age in a particular management unit is some specified value, one may wish to know whether average yield of a crop in a certain district is equal to a specified value, one may wish to compare two species of trees with regard to mean height, to know if genetic fraction of total variation in a strain is more than a given value. In making such comparisons, one can not rely on mere numerical magnitudes of index of comparison such as mean and variance. This is because each group is represented only by a sample of observations and if another sample were drawn, the numerical value would change. This variation between samples from the same population can at best be reduced in a well-designed controlled experiment but can never be eliminated. One is forced to draw inferences in presence of sampling fluctuations which affect observed differences between groups, clouding real differences. Statistical science provides an objective procedure for distinguishing whether observed difference connotes any real difference among groups. Such a procedure is called **testing of hypothesis**. Thus, in short, testing of hypothesis is a method of making due allowance for sampling fluctuation affecting results of experiments or observations. These tests have wide applications in agriculture, forestry, medicine, industry, social sciences, etc.

1.1 Definitions

Statistical Hypothesis: It is an assumption either about the form or about the parameters of a distribution. For example, average height of a particular species of tree is 50 feet, normal distribution has mean 20.

If all the parameters are completely specified, hypothesis is called a **simple hypothesis**, otherwise it is a **composite hypothesis**. For example, average height of tree is 50 feet is a simple hypothesis and average height of tree is greater than 50 feet is a composite hypothesis.

Null Hypothesis (H_0): The hypothesis under test for a sample study is called Null hypothesis (H_0). It represents a theory that has been put forward, either because it is believed to be true or because it is to be used as a basis for argument, but has not been proved. For example, in a clinical trial of a new drug, null hypothesis might be that the new drug is, on average, as effective as the current drug i.e. H_0: Effect of the two drugs, on the average, is same.

Alternative Hypothesis (H_1): Any null hypothesis is tested against a rival, which is called Alternative hypothesis (H_1). For example, mean height (μ) of trees of a particular species in a region is some specified value μ_0, i.e.

H_0: $\mu = \mu_0$.

Alternative hypothesis could be any of the following:

H_1: $\mu \neq \mu_0$ (Two-tailed)
$\mu < \mu_0$ (Left-tailed)
$\mu > \mu_0$ (Right-tailed)

For framing a suitable H_0 and H_1, four possibilities in order of preference are the following:

Possibilities	**H_0**	**H_1**
(i)	Simple	Simple
(ii)	Simple	Composite
(iii)	Composite	Simple
(iv)	Composite	Composite

The first one when both are simple is of little practical importance. As Possibility (ii) is preferred over Possibility (iii), therefore hypotheses should always be structured in such a way that H_0 is simple and H_1 is composite.

Two Types of Errors

True Situation → Decision Made ↓	**H_0 is True**	**H_0 is False**
Reject H_0	Type I error	Correct decision
Accept H_0	Correct decision	Type II error

Probabilities of these types of error are respectively denoted by α and β, i.e.

Probability of Type I error = α
and Probability of Type II error = β.

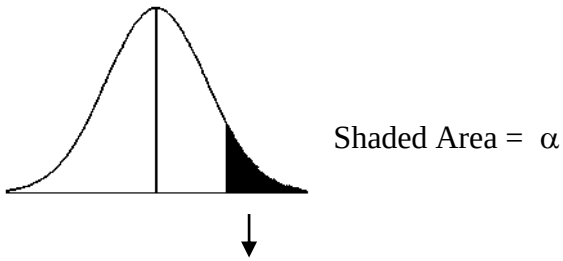
The ideal procedure of hypothesis testing is to minimize both α and β. However, this is not possible in practice because a test which minimizes one type of error, maximizes the other type of error. As Type I error is considered to be more serious than Type II error, therefore probability of Type I error is fixed and probability of Type II error is minimized. Generally, α is taken to be 5% or 1%.

Level of Significance (α): It is the size of Type I error. The higher the value of α, less precise is the result.

Confidence Interval: The confidence interval of a parameter with confidence coefficient 100(1-α)% is the interval (a, b) such that it is expected to lie in this interval in 100(1-α)% cases.

Test Statistic: A test statistic is a quantity calculated from data. Its value is used to decide whether or not the null hypothesis should be rejected.

Critical Value(s): The critical value(s) is that value with which value of test statistic in a sample is compared to determine whether or not the null hypothesis is rejected. The critical value for any hypothesis test depends on significance level α at which the test is carried out, and whether the test is one-sided or two-sided.



Power of a Test: It is defined as the probability of rejecting H_0 when it is false. Thus,

Power = 1 - β

Among a given set of tests, best test is one having maximum power.

Steps in Hypothesis Testing

- State statistical hypotheses
- Check assumptions
- Calculate test statistic
- Set the test criteria
- Interpret the results

We now discuss some tests of hypothesis that are based on normal, t, F and chi-square distributions.

2. Test of Significance for Large Samples

For large n (sample size), almost all the distributions can be approximated very closely by a normal probability curve, we therefore use the **normal test** of significance for large samples. If t is any statistic (function of sample values), then for large sample

$$Z = \frac{t - E(t)}{\sqrt{V(t)}} = N(0.1)$$

Thus if the discrepancy between the observed and the expected (hypothetical) value of a statistic is greater than Z_α times the standard error (S.E), hypothesis is rejected at α level of significance. Similarly if

$$| t - E(t) | \leq Z_\alpha \times S.E(t),$$

the deviation is not regarded significant at 5% level of significance. In other words the deviation t - E(t), could have arisen due to fluctuations of sampling and the data do not provide any evidence against the null hypothesis which may, therefore be accepted at α level of significance.

If $|Z| \leq 1.96$, then the hypothesis H_0 is accepted at 5% level of significance. Thus the steps to be used in the normal test are as follows:

i) Compute the test statistic Z under H_0.
ii) If $|Z| > 3$, H_0 is always rejected
iii) If $|Z| < 3$, we test its significance at certain level of significance

The table below gives some critical values of Z:

Level of Significance	**Critical Value (Z_α) of Z**	
	Two-tailed test	**Single tailed test**
10%	1.645	1.280
5%	1.960	1.645
1%	2.580	2.330

2.1 Test for Single Mean

A very important assumption underlying the tests of significance for variables is that the sample mean is asymptotically normally distributed even if the parent population from which the sample is drawn is not normal.

If x_i (i =1,…,n) is a random sample of size n from a normal population with mean μ and variance σ^2, then the sample mean is distributed normally with mean μ and variance $\frac{\sigma^2}{n}$. Based on this random sample, our aim is to test that mean of the population has a specified value μ_0, i.e.

H_0: $\mu = \mu_0$

The alternative hypothesis could be any of the following:

H_1: $\mu \neq \mu_0$ (two tailed)
$\mu < \mu_0$ (left tailed)
$\mu > \mu_0$ (right tailed)

Test Statistic:

$$Z = \frac{\bar{x} - \mu_0}{\sigma / \sqrt{n}}$$

follows a standard normal distribution.

Test Criteria: Depending on the alternative hypothesis selected, the test criteria are as follows:

H_1	Test	Reject H_0 at level of significance α if
$\mu \neq \mu_0$	Two-tailed	$\lvert Z \rvert > Z_{\alpha/2}$
$\mu < \mu_0$	Left-tailed	$Z < -Z_{\alpha}$
$\mu > \mu_0$	Right-tailed	$Z > Z_{\alpha}$

Z_α is the table value of Z at level of significance α. If σ^2 is unknown, then it is estimated by sample variance s^2 (for large n), where $s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$

Example 2.1: The mean timber yield obtained from 30 recently felled plantations at the age of 50 years in a particular management unit is 93 m^3/ha with a standard deviation of 10 m^3/ha. Test whether the mean timber yield is 100 m^3/ha based on past records.

Solution: H_0 : $\mu = 100$ m^3/ha, H_1: $\mu \neq 100$ m^3/ha (two tailed test).

Here, $\bar{x} = 93$ m^3/ha., n = 30, $\mu = 100$ m^3/ha and $\sigma = 10$ m^3/ha.

Thus,

$$Z = \frac{93-100}{10/\sqrt{30}} = -3.834$$

Since $|Z| > 1.96$, we conclude that the data does not provide any evidence in favour of the null hypothesis H_0 may therefore be rejected at 5% level of significance. Hence the decision would be to accept the alternative hypothesis that there has been significant decline in the productivity of the management unit with respect to the plantations of the species considered.

Note: The value of sample mean is an acceptable value of population mean if the statistic Z lies between $-Z_{\alpha/2}$ to $Z_{\alpha/2}$, i.e.

$$-Z_{\alpha/2} \le \frac{\bar{x}-\mu}{\sigma/\sqrt{n}} \le Z_{\alpha/2}.$$

Thus, 100(1-α)% confidence-interval for μ is

$$(\bar{x} - Z_{\alpha/2}\,\sigma/\sqrt{n},\ \bar{x} + Z_{\alpha/2}\,\sigma/\sqrt{n}).$$

2.2 Test for Difference of Means

Let $\bar{x}_1\,(\bar{x}_2)$ be the mean of a sample of size n_1 (n_2) from a population with mean μ_1 (μ_2) and variance $\sigma_1^2\,(\sigma_2^2)$. Our aim is to test

$H_0: \mu_1 = \mu_2$

against $H_1: \mu_1 \neq \mu_2$

$\mu_1 > \mu_2$

$\mu_1 < \mu_2$

Test Statistic:

$$Z = \frac{(\bar{x}_1 - \bar{x}_2)}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}} \sim N(0,1)$$

follows a standard normal distribution

Test Criteria:

H_1	Test	Reject H_0 at level of significance α if
$\mu_1 \neq \mu_2$	Two-tailed	$\|Z\| > Z_{\alpha/2}$
$\mu_1 < \mu_2$	Left-tailed	$Z < -Z_{\alpha}$
$\mu_1 > \mu_2$	Right-tailed	$Z > Z_{\alpha}$

$$Z = \frac{(\bar{x}_1 - \bar{x}_2)}{\sigma\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}, \text{ If } \sigma_1^2 = \sigma_2^2 = \sigma^2$$

If σ is not known, then its estimate is used

$$\hat{\sigma^2} = s^2 = \frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}$$

2.3 Test for Single Proportion

Suppose in a sample of size n (>30), x be the number of successes. Then observed proportion of successes = $x/n = p$. Let P be the population proportion. The hypothesis to be tested is that population proportion is some specified value P_0, i.e.

H_0: $P = P_0$

H_1: $P \neq P_0$

$P > P_0$

$P < P_0$

Test Statistic:

$$Z = \frac{p - P_0}{\sqrt{P_0(1 - P_0)/n}}$$

follows approximately a standard normal distribution.

Test Criteria:

H_1	Test	Reject H_0 at level of significance α if
$P \neq P_0$	Two-tailed	$\lvert Z \rvert > Z_{\alpha/2}$
$P < P_0$	Left-tailed	$Z < -Z_\alpha$
$P > P_0$	Right-tailed	$Z > Z_\alpha$

Example 2.2: In a sample of 1000 people, 540 are rice eaters and the rest are wheat eaters. Can we assume that both rice and wheat are equally popular at 1% level of significance?

Solution: It is given that n = 1000, x = Number of rice eaters = 540, p = sample proportion of rice eaters = $540/1000 = 0.54$.

H_0 : Both rice and wheat are equally popular, i.e. P = 0.5

H_1 : $P \neq 0.5$

$$Z = \frac{p - P_0}{\sqrt{P_0(1 - P_0)/n}} = \frac{0.54 - 0.5}{\sqrt{0.5 \times 0.5/1000}} = 2.532$$

Tabulated value of Z at 1% level of significance is 2.575. Since $\lvert Z \rvert < 2.575$, therefore H_0 is not rejected and we conclude that rice and wheat are equally popular.

2.4 Test for Difference of Proportions

Suppose we want to compare two populations with respect to the prevalence of a certain attribute A. Let x_1 (x_2) be the number of persons possessing the given attribute A in random sample of size n_1 (n_2) from 1st (2nd) population. Then sample proportions will be

$$p_1 = \frac{x_1}{n_1},\ p_2 = \frac{x_2}{n_2}$$

Let P_1 and P_2 be the population proportions. Our aim here is to test that there is no significant difference between population proportions, i.e.

H_0: $P_1 = P_2$

H_1: $P_1 \neq P_2$

$P_1 > P_2$

$P_1 < P_2$

Test Statistic:

$$Z = \frac{p_1 - p_2}{\sqrt{\left(\frac{P_1Q_1}{n_1} + \frac{P_2Q_2}{n_2}\right)}}$$

follows approximately a standard normal distribution. In case $P_1 = P_2 = P$ (say) and P is not known, it is estimated as follows:

$$\hat{P} = \frac{n_1p_1 + n_2p_2}{n_1 + n_2}$$

Test Criteria:

H_1	Test	Reject H_0 at level of significance α if
$P_1 \neq P_2$	Two-tailed	$\lvert Z \rvert > Z_{\alpha/2}$
$P_1 < P_2$	Left-tailed	$Z < -Z_{\alpha}$
$P_1 > P_2$	Right-tailed	$Z > Z_{\alpha}$

Consider an experiment on rooting of stem cuttings of *Casuarina equisetifolia* wherein the effect of dipping the cuttings in solutions of IBA at two different concentrations was observed. Two batches of 30 cuttings each, were subjected dipping treatment at concentrations of 50 and 100 ppm of IBA solutions respectively. Based on the observations on number of cuttings rooted in each batch of 30 cuttings, the following proportions of rooted cuttings under each concentration were obtained. At 50 ppm, the proportion of rooted cuttings was 0.5 and at 100 ppm, the proportion was 0.37. Test whether the observed

proportions are indicative of significant differences in the effect of IBA at the two concentrations.

Here, $p_1 = 0.5$ and $p_2 = 0.37$. Then $q_1 = 0.5$, $q_2 = 0.63$. The value of $n_1 = n_2 = 30$. Thus,

$$Z = \frac{0.5 - 0.37}{\sqrt{\frac{(0.5)(0.5)}{30} + \frac{(0.37)(0.63)}{30}}} = 1.024$$

Since the calculated value of Z (1.024) is less than the table value (1.96) at 5% level of significance, we can conclude that there is no significant difference between proportion rooted cuttings under the two concentration levels.

3. Test of Significance for Small Samples

In this section, the statistical tests based on t, χ^2 and F are given.

3.1 Tests Based on t-Distribution

3.1.1 Test for an Assumed Population Mean

Suppose a random sample $x_1,..,x_n$ of size n ($n \geq 2$) has been drawn from a normal population whose variance σ^2 is unknown. On the basis of this random sample the aim is to test

$H_0 : \quad \mu = \mu_0$

$H_0 : \quad \mu \neq \mu_0$

$\mu > \mu_0$

$\mu < \mu_0$

Test statistic:

$$t = \frac{\bar{x} - \mu_0}{s/\sqrt{n}} \sim t_{n-1},$$

where $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ and $s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$

The table giving the value of t required for significance at various levels of probability and for different degrees of freedom are called the t – tables which are given in Statistical Tables by Fisher and Yates. The computed value is compared with the tabulated value at α percent level of significance and at (n-1) degrees of freedom and accordingly the null hypothesis is accepted or rejected.

3.1.2 Test for the Difference of Two Population Means

Let $\bar{x}_1 (\bar{x}_2)$ be the sample mean of a sample of size n_1 (n_2) from a population with mean μ_1 (μ_2) and variance of the two population be same σ^2, which is unknown. Our aim is to test

H_0 : $\mu_1 = \mu_2$

H_1 : $\mu_1 \neq \mu_2$ or $\mu_1 > \mu_2$ or $\mu_1 < \mu_2$

Let s_i^2, i =1, 2 be sample variances of the two samples. Then common unknown population variance σ^2 is estimated as

$$s^2 = \frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}$$

Test Statistic:

$$t = \frac{\bar{x}_1 - \bar{x}_2}{s\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

which follows a t-distribution with $n_1 + n_2 - 2$ d.f.

Test Criteria:

H_1	Test	Reject H_0 at level of significance α if
$\mu_1 \neq \mu_2$	Two-tailed	$\lvert t \rvert > t_{n_1+n_2-2}(\alpha/2)$
$\mu_1 < \mu_2$	Left-tailed	$t < -t_{n_1+n_2-2}(\alpha)$
$\mu_1 > \mu_2$	Right-tailed	$t > t_{n_1+n_2-2}(\alpha)$

This test statistic is used under certain assumptions *viz*., (i) The variables involved are continuous (ii) The population from which the samples are drawn follow normal distribution (iii) The samples are drawn independently (iv) The variances of the two populations from which the samples are drawn are homogeneous (equal). The homogeneity of two variances can be tested by using F-test.

Example 3.1: A group of 5 plots treated with nitrogen at 20 kg/ha. yielded 42, 39, 48, 60 and 41 kg whereas second group of 7 plots treated with nitrogen at 40 kg/ha. yielded 38, 42, 56, 64, 68, 69 and 62 kg. Can it be concluded that nitrogen at level 40 kg/ha. increases the yield significantly?

Solution: H_0: $\mu_1 = \mu_2$, H_1: $\mu_1 < \mu_2$

Here, $\bar{x}_1 = 46$, $\bar{x}_2 = 57$, $s^2 = 121.6$

$$t = \frac{46 - 57}{\sqrt{121.6(\frac{1}{5} + \frac{1}{7})}} = -1.7 \sim t_{10}$$

Since |t| < 1.81 (value of t at 5% and 10 d.f), the yield from two doses of nitrogen do not differ significantly.

3.1.3 Paired t-test for Difference of Means

When the two samples are not independent but the sample observations are paired together, then this test is applied. The paired observations are on the same unit or matching units. For example, to know the impact of a new teaching method on the performance of students, the observations, in terms of marks, are collected before and after the new teaching method is implemented. Let (x_i, y_i), i = 1,…,n be the pairs of observations and let $d_i = x_i - y_i$. Our aim is to test

$H_0 : \mu_1 = \mu_2$
$H_1 : \mu_1 \neq \mu_2$
$\mu_1 > \mu_2$
$\mu_1 < \mu_2$

Test Statistic:

$$t = \frac{\bar{d}}{s_d/\sqrt{n}}$$

follows t distribution with n-1 d.f., where $\bar{d} = \frac{1}{n}\sum_{i=1}^{n} d_i$ and $s_d^2 = \frac{1}{n-1}\sum_{i=1}^{n}(d_i - \bar{d})^2$.

Test Criteria:

H_1	Test	Reject H_0 at level of significance α if
$\mu_1 \neq \mu_2$	Two-tailed	$\lvert t \rvert > t_{n-1}(\alpha/2)$
$\mu_1 < \mu_2$	Left-tailed	$t < -t_{n-1}(\alpha)$
$\mu_1 > \mu_2$	Right-tailed	$t > t_{n-1}(\alpha)$

3.1.4 Test for Significance of Observed Correlation Coefficient

Given a random sample (x_i, y_i) , i = 1,…, n from a bivariate normal population. We want to test the null hypothesis that the population correlation coefficient is zero i.e.

$H_0 : \rho = 0$
$H_1 : \rho \neq 0$

Test Statistic:

$$t = \frac{r\sqrt{n-2}}{\sqrt{1-r^2}} \sim t_{n-2},$$

where r is the sample correlation coefficient. H_0 is rejected at level α if

$|t| > t_{n-2}(\alpha/2)$

This test can also be used for testing the significance of rank correlation coefficient.

3.2 Test of Significance Based on Chi-Square Distribution

3.2.1 Test for the Variance of a Normal Population

Let $x_1, x_2,\ldots,x_n$ $(n\geq 2)$ be a random sample from a normal population with mean μ and variance σ^2. On the basis of this sample our aim is to test

$$H_0 : \sigma^2 = \sigma_0^2$$

against $H_1 : \sigma^2 \neq \sigma_0^2$

$$\sigma^2 < \sigma_0^2$$

$$\sigma^2 > \sigma_0^2$$

Test Statistic:

$$\chi^2 = \sum_{i=1}^{n} \left(\frac{x_i - \mu}{\sigma_0} \right)^2$$

follows a chi-square distribution with n d.f. when μ is known, and

$$\chi^2 = \sum_{i=1}^{n} \left(\frac{x_i - \bar{x}}{\sigma_0} \right)^2 = \frac{(n-1)s^2}{\sigma_0^2}$$

follows a chi-square distribution with n-1 d.f. when μ is not known.

Test Criteria:

H_1	Test	Reject H_0 at level of significance α if	
		μ is known	μ is not known
$\sigma^2 \neq \sigma_0^2$	Two-tailed	$\chi^2 < \chi_n^2(1-\alpha/2)$ or $\chi^2 > \chi_n^2(\alpha/2)$	$\chi^2 < \chi_{n-1}^2(1-\alpha/2)$ or $\chi^2 > \chi_{n-1}^2(\alpha/2)$
$\sigma^2 < \sigma_0^2$	Left-tailed	$\chi^2 < \chi_n^2(1-\alpha)$	$\chi^2 < \chi_{n-1}^2(1-\alpha)$
$\sigma^2 > \sigma_0^2$	Right-tailed	$\chi^2 > \chi_n^2(\alpha)$	$\chi^2 > \chi_{n-1}^2(\alpha)$

Tables are available for χ^2 at different levels of significance and with different degrees of freedom.

3.2.2 Test for Goodness of Fit

A test of wide applicability to numerous problems of significance in frequency data is the χ^2 test of goodness of fit. It is primarily used for testing the discrepancy between the expected and the observed frequency, For instance, one may be interested in testing whether a variable like the height of trees follows normal distribution. A tree breeder may be interested to know

whether the observed segregation ratios for a character deviate significantly from the Mendelian ratios. In such situations, we want to test the agreement between the observed and theoretical frequencies. Such a test is called a test of goodness of fit.

H_0 : the fitted distribution is a good fit to the given data
H_1 : not a good fit.

Test statistic: If O_i and E_i, i =1,…,n are respectively the observed and expected frequency of i^{th} class, then the statistic

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i} \sim \chi^2_{n-r-1}$$

where r is the number of parameters estimated from the sample, n is the number of classes after pooling. H_0 is rejected at level α if calculated χ^2 > tabulated χ^2_{n-r-1} (α).

Example 3.2: In an F_2 population of chillies, 831 plants with purple and 269 with non-purple chillies were observed. Is this ratio consistent with a single factor ratio of 3:1?

Solution: On the hypothesis of a ratio of 3:1, the frequencies expected in the purple and non-purple classes are 825 and 275 respectively.

	Frequency		
	Observed (O_i)	Expected (E_i)	O_i - E_i
Purpose	831	825	6
Non-purple	269	275	-6

$$\chi^2 = \sum_{i=1}^{2} \frac{(O_i - E_i)^2}{E_i} = 0.17.$$

Here χ^2 is based on one degree of freedom. It is seen from the table that the value of 0.17 for χ^2 with 1 d.f corresponds to a level of probability which lies between 0.5 and 0.7. It is concluded that the result is non-significant.

3.2.3 Test of Independence

Another common use of the χ^2 test is in testing independence of classifications in what are known as contingency tables. When a group of individuals can be classified in two ways, the result of the classification in two ways the results of the classification can be set out as follows:

Contingency table

Class	**A_1**	**A_2**	**A_3**
B_1	n_{11}	n_{21}	n_{31}
B_2	n_{12}	n_{22}	n_{32}
B_3	n_{13}	n_{23}	n_{33}

Such a table giving the simultaneous classification of a body of data in two different ways is called contingency table. If there are r rows and c columns the table is said to be an r x c table.

H_0: the attributes are independent

H_1: they are not independent

Test statistic:

$$\chi^2 = \sum_{j=1}^{c} \sum_{i=1}^{r} \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \sim \chi^2_{(r-1)(c-1)}$$

H_0 is rejected at level α if $\chi^2 > \chi^2_{(r-1)(c-1)}$

3.3 Test of Significance Based on F-Distribution

3.3.1 Test for the Comparison of Two Population Variances

Let x_i , i = 1,…,n_1 and x_j, j=1,…,n_2 be the two random samples of sizes n_1 and n_2 drawn from two independent normal populations $N(\mu_1, \sigma_1^2)$ and $N(\mu_1, \sigma_2^2)$ respectively. s_1^2 and s_2^2 are the sample variances of the two samples.

$$s_1^2 = \frac{1}{n_1 - 1}\sum_{i=1}^{n1}(x_i - \bar{x}_1)^2 \text{ and } s_2^2 = \frac{1}{n_2 - 1}\sum_{j=1}^{n2}(x_j - \bar{x}_2)^2$$

$$\bar{x}_1 = \frac{1}{n_1}\sum_{i=1}^{n1} x_i, \quad \bar{x}_2 = \frac{1}{n_2}\sum_{j=1}^{n2} x_j$$

$H_0 : \sigma_1^2 = \sigma_2^2$

Test statistic: Assuming $s_1^2 > s_2^2$

$$F = \frac{s_1^2}{s_2^2} \sim F_{n_1-1, n_2-1}$$

Tables are available giving the values of F required for significance at different levels of probability and for different degrees of freedom. The computed value of F is compared with the tabulated value and the inference is drawn accordingly.

3.3.2 Test for Homogeneity of Several Population Means

The test of significance based on t-distribution is an adequate procedure only for testing the significance of the difference between two sample means. In a situation when we have three or more samples to consider at a time, an alternative procedure is needed for testing the hypothesis that all the samples are drawn from the same population i.e. they have the same mean. For Example, 5 fertilizers are applied to four plots each of wheat and yield of wheat on each of the plot is obtained. The interest is to find whether effects of these fertilizers on the yields is significantly different or in other words, whether the samples have come from the

same normal population. This is done through F-test that uses the technique of Analysis of Variance (ANOVA).

ANOVA is the technique of partitioning the total variability into different known components. It consist in the estimation of the amount of variation due to each of the independent factors (causes) separately and then comparing these estimates due to assignable factors with the estimate due to chance factor or experimental error. The F statistic used for testing the hypothesis H_0: $\mu_1 = \mu_2 = \ldots = \mu_k$ $(k>2)$ is

$$F = \frac{\text{Variationa among the sample means}}{\text{Variation within the samples}}$$

Testing of Hypothesis and Analysis of Experimental Data Using R (Practical)

We shall use the following data for practical exercise.

Table 1: Experimental data 1

A	B	Treatment	C	Replication	y	x1	x2
1	0	1	1	1	200.63	16.25	1437.03
1	1	2	1	1	210.26	19.32	257.79
2	0	3	1	1	220.37	20.11	985.84
2	1	4	1	1	232.56	22.45	235.14
3	0	5	1	1	274.53	20.14	2703.29
3	1	6	1	1	285.65	29.66	2395.38
4	0	7	1	1	305.21	25.44	805.24
4	1	8	1	1	318.25	30.11	12.61
5	0	9	1	1	326.21	29.33	482.57
5	1	10	1	1	354.22	32.89	1533.83
6	0	11	1	1	389.46	28.54	1737.38
6	1	12	1	1	408.25	35.99	2992.04
1	0	1	1	2	205.32	16.45	788.46
1	1	2	1	2	210.35	19.45	1456.53
2	0	3	1	2	242.15	21.24	1038.41
2	1	4	1	2	235.15	24.88	3997.30
3	0	5	1	2	276.24	21.56	2136.88
3	1	6	1	2	286.98	28.47	1709.14
4	0	7	1	2	305.89	25.76	230.07
4	1	8	1	2	320.99	30.89	1097.22
5	0	9	1	2	324.46	29.44	50.80
5	1	10	1	2	356.23	33.45	1571.24
6	0	11	1	2	384.51	29.34	570.67
6	1	12	1	2	406.32	36.48	2941.11
1	0	1	1	3	209.46	16.82	1112.74
1	1	2	1	3	212.35	20.25	857.16
2	0	3	1	3	245.13	22.56	1012.13
2	1	4	1	3	245.69	24.56	2116.22
3	0	5	1	3	275.12	22.20	2420.09
3	1	6	1	3	285.36	30.25	2052.26
4	0	7	1	3	307.12	26.58	517.66
4	1	8	1	3	318.47	31.28	554.92
5	0	9	1	3	325.89	28.78	266.69

5	1	10	1	3	354.36	27.56	1552.53
6	0	11	1	3	385.21	28.48	1154.03
6	1	12	1	3	410.30	36.59	2966.58
1	0	1	2	1	180.63	151.30	279.77
1	1	2	2	1	190.89	197.80	2526.36
2	0	3	2	1	200.69	203.00	446.91
2	1	4	2	1	212.32	215.90	509.46
3	0	5	2	1	254.37	235.70	1042.70
3	1	6	2	1	265.21	289.40	1657.35
4	0	7	2	1	287.32	300.20	1265.38
4	1	8	2	1	298.25	324.10	1425.13
5	0	9	2	1	309.20	334.60	444.48
5	1	10	2	1	334.16	367.40	1234.54
6	0	11	2	1	364.25	398.80	1204.76
6	1	12	2	1	389.54	410.20	1056.76
1	0	1	2	2	185.23	153.20	127.17
1	1	2	2	2	190.46	195.20	2010.78
2	0	3	2	2	202.48	204.00	1222.44
2	1	4	2	2	215.35	218.70	3474.77
3	0	5	2	2	256.15	240.10	5419.45
3	1	6	2	2	264.39	284.10	1126.48
4	0	7	2	2	288.45	301.90	2709.71
4	1	8	2	2	298.15	326.10	3632.18
5	0	9	2	2	315.24	338.40	1435.03
5	1	10	2	2	336.25	366.00	2107.45
6	0	11	2	2	365.78	399.10	2472.92
6	1	12	2	2	390.68	412.60	2648.27
1	0	1	2	3	189.46	150.00	203.47
1	1	2	2	3	192.35	195.40	2268.57
2	0	3	2	3	204.57	206.50	834.67
2	1	4	2	3	234.22	219.40	1992.12
3	0	5	2	3	254.22	238.10	3231.07
3	1	6	2	3	266.49	287.30	1391.91
4	0	7	2	3	289.55	37.40	1987.55
4	1	8	2	3	301.49	328.60	2528.65
5	0	9	2	3	318.24	334.10	939.75
5	1	10	2	3	337.46	369.80	1671.00
6	0	11	2	3	363.26	393.50	1838.84
6	1	12	2	3	390.26	415.70	1852.52

1.1 Steps to be followed for testing of hypothesis

1. Start RStudio.
2. Set working directory by typing and running the following code in the editor

```
setwd(path)

#Here path is path to the directory, for example "D:/project1/data/"
```

3. Import the data running the following code

```
d1 = read_xlsx("expdata.xlsx", "Sheet1")
```

4. See the structure of the data with the following code

```
str(d1)
```

```
Classes ‘tbl_df’, ‘tbl’ and 'data.frame':    72 obs. of  8 variables:
 $ A          : num  1 1 2 2 3 3 4 4 5 5 ...
 $ B          : num  0 1 0 1 0 1 0 1 0 1 ...
 $ Treatment  : num  1 2 3 4 5 6 7 8 9 10 ...
 $ C          : num  1 1 1 1 1 1 1 1 1 1 ...
 $ Replication: num  1 1 1 1 1 1 1 1 1 1 ...
 $ y          : num  201 210 220 233 275 ...
 $ x1         : num  16.2 19.3 20.1 22.4 20.1 ...
 $ x2         : num  1437 258 986 235 2703 ...
```

5. Convert A, B, C, Treatment into factors using the code

```
d1 = within(d1,
       {
         A = factor(A)
         B = factor(B)
         C = factor(C)
         Treatment = factor(Treatment)
         Replication = factor(Replication)
       })
```

6. Perform one sample t-test. For example, to test that population mean of y is equal to 20 versus alternative that the mean is not 20, use the code

```
t.test(d1$y, mu = 20)
```

Output:

```
        One Sample t-test

data:  d1$y
t = 34.289, df = 71, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 20
95 percent confidence interval:
 270.5266 301.4622
sample estimates:
mean of x
 285.9944
```

7. Perform two independent sample t-test. For example, to test that population mean of y is equal for the two levels of B, use the code:

```
t.test(y ~ B, data = d1, var.equal = TRUE)
```

```
        Two Sample t-test

data:  y by B
t = -0.94396, df = 70, p-value = 0.3484
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -45.62351  16.31051
sample estimates:
mean in group 0 mean in group 1
       278.6661        293.3226
```

2. R for analysis of experimental data

We shall use the data in Table 1 to analyze data using R.

Install the following packages in RStudio.

```
agricolae
emmeans
multcompView
```

2.1 Steps for analysis from CRD

The data in Table 1 is from an experiment using a completely randomized design with 12 treatments.

1. Follow steps 1 to 6 of Section 1.

2. Write the following codes in editor and run them.

```
result1 = lm(y ~ Treatment, data = d1)
anova(result1)
```

Output:

```
Analysis of Variance Table

Response: y
          Df Sum Sq Mean Sq F value    Pr(>F)
Treatment 11 299123 27193.0   191.9 < 2.2e-16 ***
Residuals 60   8502   141.7
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

3. Write the following code in editor and run it.

```
LSD.test(result1, "Treatment", console = TRUE)
```

Output:

```
Study: result1 ~ "Treatment"

LSD t Test for y

Mean Square Error:  141.7056

Treatment,  means and individual ( 95 %) CI

           y       std r      LCL      UCL     Min     Max
1  195.1217 11.660077 6 185.4006 204.8427 180.633 209.458
10 345.4442 10.472961 6 335.7231 355.1652 334.158 356.231
11 375.4112 12.175071 6 365.6901 385.1322 363.255 389.456
12 399.2232 10.015863 6 389.5021 408.9442 389.540 410.298
2  201.1103 10.865020 6 191.3893 210.8314 190.457 212.354
3  219.2308 20.182929 6 209.5098 228.9519 200.689 245.126
4  229.2148 12.810761 6 219.4938 238.9359 212.321 245.690
5  265.1033 11.199877 6 255.3823 274.8244 254.215 276.235
6  275.6783 11.333499 6 265.9573 285.3994 264.389 286.976
7  297.2572  9.704150 6 287.5361 306.9782 287.320 307.124
8  309.2650 11.028341 6 299.5440 318.9860 298.147 320.986
9  319.8727  6.861633 6 310.1516 329.5937 309.200 326.210

Alpha: 0.05 ; DF Error: 60
```

```
Critical Value of t: 2.000298

least Significant Difference: 13.74762

Treatments with the same letter are not significantly different.

          y groups
12 399.2232      a
11 375.4112      b
10 345.4442      c
9  319.8727      d
8  309.2650     de
7  297.2572      e
6  275.6783      f
5  265.1033      f
4  229.2148      g
3  219.2308      g
2  201.1103      h
1  195.1217      h
```

4. Write the following code in editor and run.

```
cv.model(result1)
```

Output:

```
[1] 4.162325
```

2.2 Steps for analysis from RCBD

The data in Table 1 is from an experiment using a randomized complete block design with 12 treatments and 3 blocks. Within each block, 12 treatments appear once.

1. Follow steps 1 to 6 of Section 1.
2. Write the following codes in editor and run them.

```
result2 = lm(y ~ Replication + Treatment, data = d1)

anova(result2)
```

Output

```
Analysis of Variance Table

Response: y
```

```
            Df Sum Sq Mean Sq  F value Pr(>F)
Replication  2    223   111.7   0.7829 0.4619
Treatment   11 299123 27193.0 190.5089 <2e-16 ***
Residuals   58   8279   142.7
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

3. Write the following code in editor and run.

```
LSD.test(result2, "Treatment", console = TRUE)
```

Output

```
Study: result2 ~ "Treatment"

LSD t Test for y

Mean Square Error:  142.7386

Treatment,  means and individual ( 95 %) CI

           y       std r      LCL      UCL     Min     Max
1  195.1217 11.660077 6 185.3583 204.8850 180.633 209.458
10 345.4442 10.472961 6 335.6808 355.2075 334.158 356.231
11 375.4112 12.175071 6 365.6478 385.1745 363.255 389.456
12 399.2232 10.015863 6 389.4598 408.9865 389.540 410.298
2  201.1103 10.865020 6 191.3470 210.8737 190.457 212.354
3  219.2308 20.182929 6 209.4675 228.9942 200.689 245.126
4  229.2148 12.810761 6 219.4515 238.9782 212.321 245.690
5  265.1033 11.199877 6 255.3400 274.8667 254.215 276.235
6  275.6783 11.333499 6 265.9150 285.4417 264.389 286.976
7  297.2572  9.704150 6 287.4938 307.0205 287.320 307.124
8  309.2650 11.028341 6 299.5017 319.0283 298.147 320.986
9  319.8727  6.861633 6 310.1093 329.6360 309.200 326.210

Alpha: 0.05 ; DF Error: 58
Critical Value of t: 2.001717

least Significant Difference: 13.80743

Treatments with the same letter are not significantly different.

           y groups
12 399.2232      a
11 375.4112      b
10 345.4442      c
```

```
9  319.8727      d
8  309.2650     de
7  297.2572      e
6  275.6783      f
5  265.1033      f
4  229.2148      g
3  219.2308      g
2  201.1103      h
1  195.1217      h
```

4. Write the following code in editor and run.

```
cv.model(result2)
```

Output

```
[1] 4.177469
```

2.3 Steps for analysis from factorial experiments

The data in Table 1 is from an experiment with a factorial experiment with two factors A and B with 6 levels of factor A and 2 levels of factor B. Design is completely randomized design.

1. Follow steps 1 to 6 of Section 1.
2. Write the following codes in editor and run them.

```
result3 = lm(y ~ A + B + A:B, data = d1)
anova(result3)
```

Output:

```
Analysis of Variance Table

Response: y
          Df Sum Sq Mean Sq  F value    Pr(>F)
A          5 294285   58857 415.3476 < 2.2e-16 ***
B          1   3867    3867  27.2864  2.32e-06 ***
A:B        5    971     194   1.3702    0.2483
Residuals 60   8502     142
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

3. Write the following code in editor and run.

```
LSD.test(result3, "A", console = TRUE)
```

Output:

```
Study: result3 ~ "A"

LSD t Test for y

Mean Square Error:  141.7056

A,  means and individual ( 95 %) CI

         y      std  r      LCL      UCL     Min     Max
1 198.1160 11.19100 12 191.2422 204.9898 180.633 212.354
2 224.2228 16.93939 12 217.3490 231.0966 200.689 245.690
3 270.3908 12.07898 12 263.5170 277.2646 254.215 286.976
4 303.2611 11.72231 12 296.3873 310.1349 287.320 320.986
5 332.6584 15.79853 12 325.7846 339.5322 309.200 356.231
6 387.3172 16.35899 12 380.4434 394.1910 363.255 410.298

Alpha: 0.05 ; DF Error: 60
Critical Value of t: 2.000298

least Significant Difference: 9.721036

Treatments with the same letter are not significantly different.

         y groups
6 387.3172      a
5 332.6584      b
4 303.2611      c
3 270.3908      d
2 224.2228      e
1 198.1160      f
```

4. Write the following code in editor and run.

```
LSD.test(result3, "B", console = TRUE)
```

Output:

```
Study: result3 ~ "B"

LSD t Test for y
```

```
Mean Square Error:  141.7056

B,  means and individual ( 95 %) CI

         y      std  r      LCL      UCL     Min     Max
0 278.6661 62.65267 36 274.6975 282.6347 180.633 389.456
1 293.3226 68.94533 36 289.3540 297.2912 190.457 410.298

Alpha: 0.05 ; DF Error: 60
Critical Value of t: 2.000298

least Significant Difference: 5.612443

Treatments with the same letter are not significantly different.

         y groups
1 293.3226      a
0 278.6661      b
```

5. Write the following code in editor and run.

```
CLD(emmeans(result3, ~ A*B), Letters = letters)
```

Output:

```
A B   emmean       SE df lower.CL upper.CL .group
 1 0 195.1217 4.859794 60 185.4006 204.8427  a
 1 1 201.1103 4.859794 60 191.3893 210.8314  ab
 2 0 219.2308 4.859794 60 209.5098 228.9519   bc
 2 1 229.2148 4.859794 60 219.4938 238.9359    c
 3 0 265.1033 4.859794 60 255.3823 274.8244     d
 3 1 275.6783 4.859794 60 265.9573 285.3994     de
 4 0 297.2572 4.859794 60 287.5361 306.9782      ef
 4 1 309.2650 4.859794 60 299.5440 318.9860       f
 5 0 319.8727 4.859794 60 310.1516 329.5937       f
 5 1 345.4442 4.859794 60 335.7231 355.1652        g
 6 0 375.4112 4.859794 60 365.6901 385.1322         h
 6 1 399.2232 4.859794 60 389.5021 408.9442          i

Confidence level used: 0.95
P value adjustment: tukey method for comparing a family of 12 estimates
significance level used: alpha = 0.05
```

6. Write the following code in editor and run.

```
cv.model(result3)
```

Output:

```
[1] 4.162325
```

Big Data Analytics for Bioinformatics

In this post genomic era after invention of Next Generation Sequencing (NGS), millions of sequences and sequence tags information are being generated everyday by researcher across globe. The groundbreaking discovery of NGS has given scientists the means to decipher and analyze billions of DNA sequences to determine what specific genes do, and gain the insight into how the body works to develop new therapeutics.

One significant obstacle is NGS analysis produces massive volumes of data; up to a terabyte for a single DNA sample. The conventional approach to assemble raw data sequences, create DNA annotations and root out false negatives and errors in variant call data is a tedious processes that greatly slow the analysis. Since the size of the data is in petabytes consisting of billions of records of millions of genes, proteins, human-genome: all from different sources (e.g. Database, wet laboratory, dry laboratory, web, scientific researches, and so on); while dealing with such a huge datasets, researchers face lots of difficulties in being able to access, create, manipulate, store, manage and analyze such a huge data. Further the major difficulty faced by such a voluminous data is particularly in business analytics because of lack of standard tools and procedures. Big Data Analytics is the process of typically applying the tools of artificial intelligence, like machine learning, to a heap of data beyond that which can be captured in standard databases.

The inter-correlation, association mining, network creation, functional annotation and extraction of meaningful information (KDD) need an efficient use of these big-data by existing tools and algorithm poses and challenging area of research. Data aggregation is one time computational intensive work which offers a smooth searching and analyzing arena through series of unsupervised and supervised algorithms.

1. Size of Data:

The size of data often can be considered as ‘Big Data’, depending on its processing, prospective and goal/objective of data mining. Though it is still widely considered with fuzzy understanding as 1TB or more. Big Data, while impossible to define specifically, may be defined as junk of data which need to be process on utility nodes which typically generate with a speed more than its inferring engine.

Hence it can be safely assumed to have three main characteristics: Velocity (speed of data in and out), Variety (range of data types and sources), and Volume (amount of data). Velocity describes the frequency at which data is generated, captured and shared. Variety of big data means much more than rows and columns, it means unstructured data that can have important impacts on company decisions, if it’s analyzed properly in time. Volume describes

the amount of data generated by organizations or individuals. Big Data is usually associated with this characteristic.

Next generation Sequencing Data with large size of the FASTQ files creates the problem for effective analysis with conventional computational tools and hardware. For example, compressed FASTQ files from a typical human whole genome sequencing can still require 800 Gb. A small project with 10 whole genome sequencing (WGS) samples can generate more than ~8 TB of raw data. The disk space required for downstream analysis will add up the secondary memory further.

2. Diversity in Data Set

The holy grail of any data analytics is the data itself; the data which can represent the overall scenario without repeating representations for an individual 'local solution space' is the best way to go through. So the variation is required in sampling with least replication (which is practically different from preassembled repeat NGS tags). In most cases when a sequence file is processed for a specific patterns; it may present in repeating fashion through many similar instances if not same. Overrepresentation often produce biased training and hence wrong prediction, which cannot be safely nullified by Cross-Validation methods. Hence the dataset sampling itself is very crucial for all pattern recognition task which relies heavily on diversity management in dataset.

3. Clustering

Clustering is the task of aggregating a set of data in such a way that individual data in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters). The aggregated data and its cluster depends on various factors like, distance function, type of clustering, data representation, cardinal mapping of attribute values etc. However we are more interested to look into those algorithms which are specific for our task of NGS data analysis; which can be summarized as Hierarchical Clustering, K-Mean and C-Fuzzy Means clustering. One centroid data can safely represent the aggregate of data in one cluster if feed with a proper distance function and clustering kernel.

4. Pattern Recognition & Classification

Pattern recognition focuses on the recognition of patterns and regularities or conserved motifs in data, which may carry the intrinsic characteristic of the data. Sequence data carries patterns as isolated islands or in association with multiple supporting motifs. Few popular algorithms in this area are

- Decision trees

- K-nearest-neighbor algorithms
- Naive Bayes classifier
- Neural networks
- Support vector machines etc.

Classification of Big Data

With limitation of processing and storage traditional analytics tools are not well suited to capturing the full meaning of big data. The volume of data is often too large for comprehensive analysis, and the requirement of establishing correlations and relationships between disparate data sources require intensive processing. Machine learning is ideal for exploiting the opportunities hidden in big data. The data analyst need to test all hypotheses and derive all the values buried within the data. Basic analytical methods used in business intelligence and enterprise reporting tools works on structured data through running SQL queries. However through systematized extension of these basic analytics on can achive similar results on big data through set of commodity hardware clustered in a scalable distributed environment.

Tools and Open Sources

a) Scikit-learn:

Scikit-learn leverages this breadth by building on top of several existing Python packages -- NumPy, SciPy, and matplotlib - for math and science work. The resulting libraries can be used either for interactive “workbench” applications or be embedded into other software and reused. The kit is available under a BSD license, so it’s fully open and reusable.

GitHub: https://github.com/scikit-learn/scikit-learn

b) Shogun

Shogun was created and written in C++, but isn’t limited to working in C++. The SWIG library, facilitate the Shogun to be used transparently with languages and environments as Java, Python, C#, Ruby, R, Lua, Octave, and Matlab.

GitHub: https://github.com/shogun-toolbox/shogun

c) Accord Framework/AForge.net

Accord, a machine learning and signal processing framework for .Net, includes libraries that provide a more conventional gamut of machine learning functions, from neural networks to decision-tree systems. Works preferably for signal processing and image analysis.

GitHub: https://github.com/accord-net/framework/

d) Mahout

The Mahout framework has long been tied to Hadoop, but many of the algorithms under its umbrella can also run as-is outside Hadoop. They're useful for stand-alone applications that runs on Hadoop or stand-alone applications. One downside of Mahout is its usage of the legacy (obsolete) MapReduce framework instead of Spark.

e) MLlib

Apache's own machine learning library for Spark and Hadoop, MLlib boasts a gamut of common algorithms and useful data types, designed to run at speed and scale. As you'd expect with any Hadoop project, Java is the primary language for working in MLlib, but Python users can connect MLlib with the NumPy library (also used in scikit-learn), and Scala users can write code against MLlib. MLlib can be deployed on top of Spark without Hadoop and in EC2 or on Mesos. Another project, MLbase, builds on top of MLlib to make it easier to derive results. Rather than write code, users make queries by way of a declarative language.

f) H2O

Oxdata's H2O's algorithms are geared for interacting in a stand-alone fashion with HDFS stores, on top of YARN, in MapReduce, or directly in an Amazon EC2 instance. Hadoop mavens can use Java to interact with H2O, but the framework also provides bindings for Python, R, and Scala, providing cross-interaction with all the libraries available on those platforms as well.

GitHub: *https://github.com/0xdata/h2o*

g) Cloudera Oryx

Yet another machine learning project designed for Hadoop, Oryx comes courtesy of the creators of the Cloudera Hadoop distribution. The name on the label isn't the only detail that sets Oryx apart: Per Cloudera's emphasis on analyzing live streaming data by way of the Spark project, Oryx is designed to allow machine learning models to be deployed on real-time streamed data, enabling projects like real-time spam filters or recommendation engines

GitHub: https://github.com/cloudera/oryx

h) GoLearn

Google's Go language has been in the wild for only five years, but has started to enjoy wider use, due to a growing collection of libraries. The simplicity comes from the way data is

loaded and handled in the library, since it's patterned after SciPy and R. The customizability lies in both the library's open source nature (it's MIT-licensed) and in how some of the data structures can be easily extended in an application.One of the libraries found in the Shogun toolbox.

GitHub: *https://github.com/sjwhitworth/golearn*

i) Weka

Weka, a product of the University of Waikato, New Zealand, collects a set of Java machine learning algorithms engineered specifically for data mining. This GNU GPLv3-licensed collection has a package system to extend its functionality, with both official and unofficial packages available. Weka even comes with a book to explain both the software and the techniques used, so those looking to get a leg up on both the concepts and the software may want to start there. While Weka isn't aimed specifically at Hadoop users, it can be used with Hadoop thanks to a set of wrappers produced for the most recent versions of Weka. Note that it doesn't yet support Spark, only MapReduc. Clojure users can also leverage Weka, thanks to the Clj-ml library.

j) CUDA-Convnet

By now most everyone knows how GPUs can crunch certain problems faster than CPUs. But applications don't automatically take advantage of GPU acceleration; they have to be specifically written to do so. CUDA-Convnet is a machine learning library for neural-network applications, written in C++ to exploit the Nvidia's CUDA GPU processing technology. For those using Python rather than C++, the resulting neural nets can be saved as Python objects and thus accessed from Python. Note that original version of the project is no longer being developed, but has since been reworked into a successor, CUDA-Convnet2, with support for multiple GPUs and Kepler-generation GPUs. A similar project, Vulpes, has been written in F# and works with the .Net framework generally.

k) ConvNetJS

As the name implies, ConvNetJS provides neural network machine learning libraries for use in JavaScript, facilitating use of the browser as a data workbench. An NPM version is also available for those using Node.js, and the library is designed to make proper use of JavaScript's asynchronicity -- for example, training operations can be given a callback to execute once they complete.

GitHub: *https://github.com/karpathy/convnetjs*

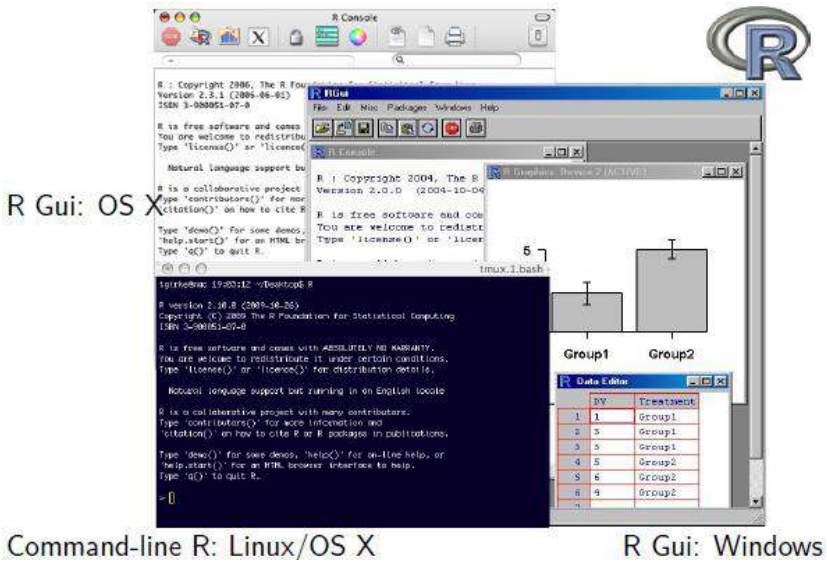
R for High Dimensional Data Analysis

(Practical)

Why R?

It's free! And runs on various platforms including Windows, Unix and Mac OSX.
It provides an unparalleled platform for developing and sharing statistical methods across the community.
R contains many advanced statistical routines not yet available in other packages.
It has state-of-the-art graphics capabilities.
Bioconductor: state-of-the-art bio-informatic modules

R: Interface



R Workflow

R is not menu-driven. Its primarily command-based.
R is an interactive language. Enter a command, wait for it to finish, enter the next command and so on.
R is typically not meant for 'standardized analysis pipelines'. Generally the output of each intermediate step should be examined.

Input-1 <- read-Data()
Output-1 <- processing-step-1 (Input-1)
Output-2 <- processing-step-2 (Output-1)
….
Output-final

The steps can be all run together using an R-script if many repetitions are needed (also to reproduce the analysis later).

Data Types

R has a wide variety of data types including

Scalars (i.e. vectors of length 1)

Vectors (numerical, character, logical) – 1D arrays

Matrices – 2D arrays

Data-frames – Matrix with different data types in different columns

Lists – An arbitrary collection of objects not necessarily of same length (possibly a mixture of vectors, matrices and scalars).

Numbers and Character Strings

```
Numeric data: 1, 2, 3
> x <- c(1, 2, 3); x
[1] 1 2 3
> is.numeric(x)
[1] TRUE
> as.character(x)
[1] "1" "2" "3"
Character data: "a", "b", "c"
> x <- c("1", "2", "3"); x
[1] "1" "2" "3"
> is.character(x)
[1] TRUE
> as.numeric(x)
[1] 1 2 3
```

Logical Type

```
Logical data
> x <- 1:10 < 5
> x
 [1]  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
> !x
 [1] FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
> which(x) # Returns index for the 'TRUE' values in logical vector
[1] 1 2 3 4
```

Vectors and Factors

Vectors (1D)

```
> myVec <- 1:10; names(myVec) <- letters[1:10]
> myVec[1:5]

a b c d e
1 2 3 4 5

> myVec[c(2,4,6,8)]

b d f h
2 4 6 8

> myVec[c("b", "d", "f")]

b d f
2 4 6
```

Factors (1D): vectors with grouping information

```
> factor(c("dog", "cat", "mouse", "dog", "dog", "cat"))

[1] dog   cat   mouse dog   dog   cat
Levels: cat dog mouse
```

Matrices, Data Frames

Matrices (2D): two dimensional structures with data of same type

```
> myMA <- matrix(1:30, 3, 10, byrow = TRUE)
> class(myMA)

[1] "matrix"

> myMA[1:2,]

     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,]    1    2    3    4    5    6    7    8    9    10
[2,]   11   12   13   14   15   16   17   18   19    20
```

Data Frames (2D): two dimensional structures with variable data types

```
> myDF <- data.frame(Col1=1:10, Col2=10:1)
> myDF[1:2, ]

  Col1 Col2
1    1   10
2    2    9
```

Data Subsetting

Subsetting by positive or negative index/position numbers

```
> myVec <- 1:26; names(myVec) <- LETTERS
> myVec[1:4]
```

```
A B C D
1 2 3 4
```

Subsetting by same length logical vectors

```
> myLog <- myVec > 10
> myVec[myLog]
```

```
 K  L  M  N  O  P  Q  R  S  T  U  V  W  X  Y  Z
11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
```

Subsetting by field names

```
> myVec[c("B", "K", "M")]
```

```
 B  K  M
 2 11 13
```

Calling a single column or list component by its name with the $ sign

```
> iris$Species[1:8]
```

```
[1] setosa setosa setosa setosa setosa setosa setosa setosa
Levels: setosa versicolor virginica
```

Lists

Lists: containers for any object type

```
> myL <- list(name="Fred", wife="Mary", no.children=3, child.ages=c(4,7,9))
> myL
```

```
$name
[1] "Fred"

$wife
[1] "Mary"

$no.children
[1] 3

$child.ages
[1] 4 7 9
```

```
> myL[[4]][1:2]
```

```
[1] 4 7
```

Functions

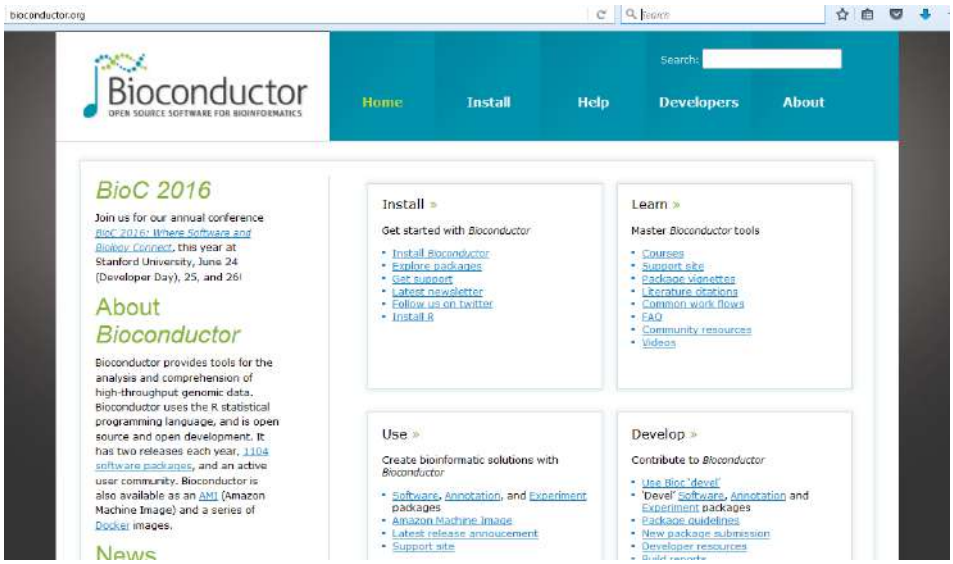
Syntax

Out1 <- processFun(obj1, arg2=val2, arg5=val5)

Generally functions consists of multiple input arguments, some with default values.
Built-in functions and help.
Most of R-s functionality is derived through functions
Examples: ,mean(), summary(), table(), plot(), hclust(), heatmap()
Type '?table' or '?heatmap' to browse the help-files.

Apart from explaining the input and output formats of these functions help-files also contain useful examples at the end.

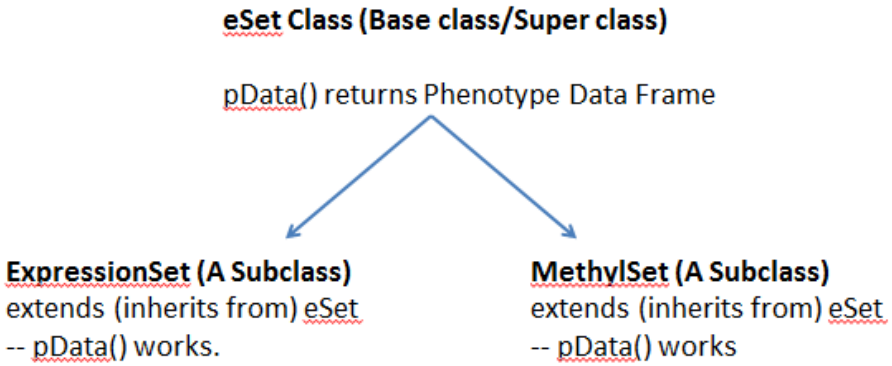
Bioconductor



Object-Oriented Model
Classes: Classes describe the general structure of any R object. "Matrix", "List" etc are all classes. Classes can be more complex. There are classes to hold, input data output from a function.
All information in a VCF formatted file ("VCF")
Raw data of a Gene-expression study ("AffyBatch").
Normalized data from a gene-expr study (e.g. "ExpressionSet")
Slots: Simpler data objects inside a class
E.g. ExpressionSet contains slots such as assayData, phenoData and featureData etc.

Methods are functions defined within a class.
An object can access methods defined by its own class.
E.g. Calling the 'pData(obj)' will return the phenoData matrix of the ExpressionSet object obj.
The biggest advantage of classes over 'lists' is that they can 'extend' other classes and 'inherit' from them.
They can inherit methods and also define methods specific to themselves.
Sometimes a method with the same name can do different things depending on the class of object it is called on.

Inheritance Example



Fewer function/method names to remember.

Bioconductor Analysis Pipelines

Read raw data into R (using a bioconductor package) as an object of a certain class.
Perform QC & Pre-processing (e.g. normalization) of the raw data to produce an object of another class.
Use appropriate method to convert normalized data to matrix or vector form.
Perform statistical analysis using CRAN/Bioconductor packages.
Visualization using R plotting functions. Sometimes convert to tracks for visualization through UCSC genome browser.

Bioconductor Help
Class introspection
getClass, getSlots, slotNames, extends
Method introspection
showMethods("exprs"),
showMethods(class="ExpressionSet")
getMethod("exprs", "ExpressionSet")

Each CRAN/BioC package has a manual with documentation of all the functions.

Almost all packages have one or more vignettes demonstrating how to use the package functions.
R repositories.
CRAN (Comprehensive R Archive Network)
More than 3500 packages
Install and 'load' a new package as follows:
install.packages("rgl")
library("rgl")
CRAN Task views (https://cran.r-project.org/web/views/).
Bioconductor
More than 1000 packages

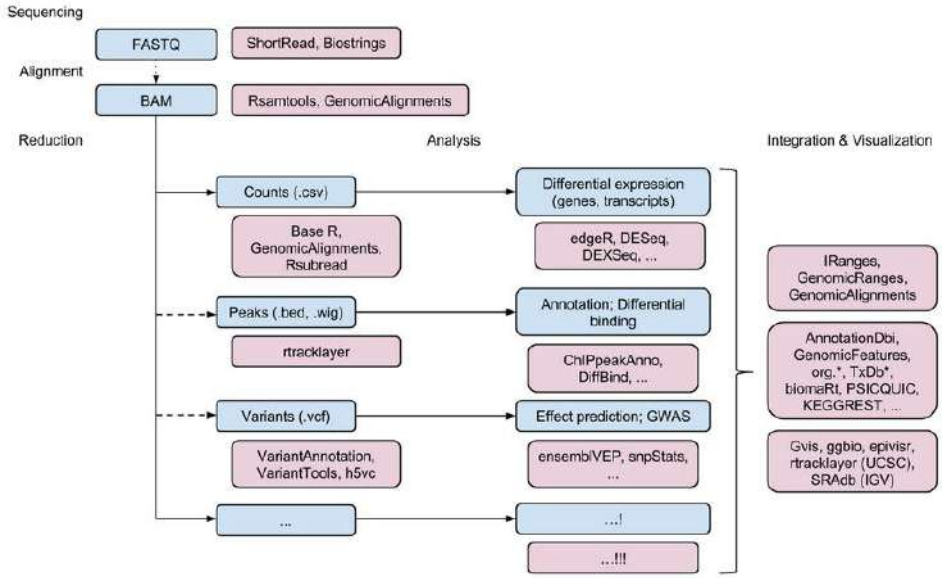
Install and 'load' a new package as follows:
source("https://bioconductor.org/biocLite.R")
biocLite("VariantAnnotation")
library("VariantAnnotation")
BiocViews (bioconductor.org/packages/release/BiocViews.html)

Example: Variant Annotation
Code to annotate variants:

```
library(TxDb.Hsapiens.UCSC.hg19.knownGene)
txdb.hg19 <- TxDb.Hsapiens.UCSC.hg19.knownGene
gr.hg19 <- rowRanges(vcf)
loc.hg19 <- locateVariants(gr.hg19, txdb.hg19, AllVariants())
table(loc.hg19$LOCATION)
## ## spliceSite    intron    fiveUTR   threeUTR    coding  intergenic  promoter
## ##        326    211920      1249      5083     11523     36207     12343
```

Similar and more complex examples can be found in Bioconductor workflows (bioconductor.org/help/workflows/).

Some Bioconductor Packages



References

https://cran.r-project.org/doc/contrib/Torfs+Brauer-Short-R-Intro.pdf
https://cran.r-project.org/doc/manuals/r-release/R-intro.pdf
https://bioconductor.org/help/course-materials/2010/SeattleIntro/IntroToR_Slides.pdf
https://master.bioconductor.org/help/course-materials/2002/Summer02Course/Labs/basics.pdf
https://bioconductor.org/help/course-materials/2010/SeattleIntro/Bioconductor-Introduction.pdf

R for Biomolecular Sequence Data

(Practical)

Packages to be installed:

```
> library(Biostrings)
> library(BSgenome)
> library(biomaRt)
> library(GenomeGraphs)
```

How would you generate a random DNA string?

```
> DNA_ALPHABET
 [1] "A" "C" "G" "T" "M" "R" "W" "S" "Y"
[10] "K" "V" "H" "D" "B" "N" "-" "+"
> seq <- sample(DNA_ALPHABET[1:4],
+     size = 24, replace = TRUE)
> seq <- DNAString(paste(seq, collapse = ""))
> seq
  24-letter "DNAString" instance
seq: TTAGTTTCACATAGCCAGCCTGCC
```

Finding out the frequency of a nucleotides in a sequence data

```
> alphabetFrequency(seq, baseOnly = T,
+     as.prob = T)
        A         C         G         T
0.2083333 0.3333333 0.1666667 0.2916667
    other
0.0000000
```

Finding out the reverse complement of a DNA sequence

```
> reverseComplement(seq)
  24-letter "DNAString" instance
seq: GGCAGGCTGGCTATGTGAAACTAA
```

Translate a nucleotide sequence data into protein sequence

```
> translate(seq)
  8-letter "AAString" instance
seq: LVSHSQPA
```

Extraction of fragment of a sequence with subset operator

```
> seq[3:10]
  8-letter "DNAString" instance
seq: AGTTTCAC
```

However, Biostrings provides the "subseq" function, that follows the SEW (Start End Width) interface. In other words, a subsequence can be extracted by two out of three possible parameters:

- start
- end
- width

```
> subseq(seq, start = 3, end = 10)
  8-letter "DNAString" instance
seq: AGTTTCAC
> subseq(seq, start = 3, width = 8)
  8-letter "DNAString" instance
seq: AGTTTCAC
> subseq(seq, end = 10, width = 8)
  8-letter "DNAString" instance
seq: AGTTTCAC
```

This function is very versatile. It even allows negative position. What does a negative position mean?

```
> subseq(seq, start = 1, end = -4)
  21-letter "DNAString" instance
seq: TTAGTTTCACATAGCCAGCCT
```

Let's create a DNAStringSet object:

```
> set <- NULL
> for (i in c(1:4)) set <- c(set,
+     paste(sample(DNA_ALPHABET[1:4],
+         30, replace = T), collapse = ""))
```

```
> set
[1] "CATGCAAATACCTTTTATTGGGGGTCAGAA"
[2] "GCTAAGCGGATTGGAGCCCTCCTCCTTTAG"
[3] "CAACCCGCATGGTAAGTTGACACCACCCGT"
[4] "TACCTTGGGTTACCCCGCGCAGCTTGCTCT"
> set <- DNAStringSet(set)
> set
  A DNAStringSet instance of length 4
    width seq
[1]    30 CATGCAAATACCTTTTATTGGGGGTCAGAA
[2]    30 GCTAAGCGGATTGGAGCCCTCCTCCTTTAG
[3]    30 CAACCCGCATGGTAAGTTGACACCACCCGT
[4]    30 TACCTTGGGTTACCCCGCGCAGCTTGCTCT
```

- The "reverseComplement", "alphabetFrequency" and "subseq" functions can be used over all the Biostrings.
- The "length" function will now return the number of sequences and "width" will return the length of each sequence.
-

```
> names(set) <- seq(4)
> set
  A DNAStringSet instance of length 4
    width seq       names
[1]    30 CAT...GAA 1
[2]    30 GCT...TAG 2
[3]    30 CAA...CGT 3
[4]    30 TAC...TCT 4
```

Reading FASTA files

- There is a special function for reading FASTA files and creating a XStringSet: read.DNAStringSet (RNA or Proteins can also be read by changing the prefix).
- The function for writing a FASTA file from an XStringSet is write.XStringSet.

Preprocessed genomes

- A package that is related to Biostrings is BSgenome.
- BSgenome provides preprocessed genomes from some model organisms, as Biostrings.

```
> available.genomes()
```

- In this session we will use the *Escherichia coli APEC 01* genome (NC_008563), so

```
> require(BSgenome.Ecoli.NCBI.20080805)
> eco <- Ecoli$NC_008563
```

Generating Views

- An object of XStringViews represents a set of subsequences from a subject string that are defined by the SEW interface.

- The views are generated by the function Views and can be defined in different ways:

```
> Views(eco, start = c(10, 20, 30,
+     40), end = c(50, 60, 70, 80))

  Views on a 5082025-letter DNAString subject
subject: AACGGGCAATATGT...TTCATTCTGACTGC
views:
    start end width
[1]    10  50    41 [TATGTCTC...ATAGCAG]
[2]    20  60    41 [TGTGGATT...CTGAACT]
[3]    30  70    41 [AAAAAGAG...TACCTGC]
[4]    40  80    41 [TCTGATAG...GAGTAAA]

> Views(eco, start = c(10, 20, 30,
+     40), end = c(50, 60))

  Views on a 5082025-letter DNAString subject
subject: AACGGGCAATATGT...TTCATTCTGACTGC
views:
    start end width
[1]    10  50    41 [TATGTCTC...ATAGCAG]
[2]    20  60    41 [TGTGGATT...CTGAACT]
[3]    30  50    21 [AAAAAGAG...ATAGCAG]
[4]    40  60    21 [TCTGATAG...CTGAACT]

> Views(eco, start = c(10, 20, 30,
+     40), width = c(100))

  Views on a 5082025-letter DNAString subject
subject: AACGGGCAATATGT...TTCATTCTGACTGC
views:
    start end width
[1]    10 109   100 [TATGTCTC...CACTAAA]
[2]    20 119   100 [TGTGGATT...TTTAACC]
[3]    30 129   100 [AAAAAGAG...ATAGGCA]
[4]    40 139   100 [TCTGATAG...CGCACAG]
```

The sliding windows

- Bioinformaticians love to use sliding windows for their analysis. Briefly, sliding windows are overlapping fragments of a sequence generated by "walking" through it.

- How would you create a set of windows of width = 100, and sliding step = 10, of the first 10kb of E. coli's genome?

```
> v1 <- Views(eco, start = seq(from = 1,
+     to = 9901, by = 10), width = 100)
> v2 <- successiveViews(eco, from = 1,
+     width = rep(100, 991), gapwidth = -90)
```

```
> head(v1)

  Views on a 5082025-letter DNAString subject
subject: AACGGGCAATATGT...TTCATTCTGACTGC
views:
    start end width
[1]     1 100   100 [AACGGGCA...GACTTAG]
[2]    11 110   100 [ATGTCTCT...ACTAAAT]
[3]    21 120   100 [GTGGATTA...TTAACCA]
[4]    31 130   100 [AAAAGAGT...TAGGCAT]
[5]    41 140   100 [CTGATAGC...GCACAGA]
[6]    51 150   100 [CTTCTGAA...ATAAAAA]
```

```
> tail(v2)

  Views on a 5082025-letter DNAString subject
subject: AACGGGCAATATGT...TTCATTCTGACTGC
views:
    start   end width
[1]  9851  9950   100 [TATATTG...GAGCGG]
[2]  9861  9960   100 [TTGCACG...AGCTTA]
[3]  9871  9970   100 [TTGTAGG...TTAGTG]
[4]  9881  9980   100 [GATAAGG...TCACCA]
[5]  9891  9990   100 [TCACGCC...GCAGAA]
[6]  9901 10000   100 [TCCGGCA...GCGACC]
```

Biostrings provide useful pattern matching functions:

- matchPattern: For matching one pattern to one string.
- vmatchPattern: For matching one pattern to several strings (StringSet).
- matchPDict: For matching a dictionary of equal length patterns to a string.
- vmatchPDict: For matching a dictionary of patterns to a collection of strings.

matchPattern

```
> motif <- DNAString("GAATTC")
> tail(matchPattern(motif, eco))

  Views on a 5082025-letter DNAString subject
subject: AACGGGCAATATGT...TTCATTCTGACTGC
views:

       start     end width
[1] 5012393 5012398     6 [GAATTC]
[2] 5047471 5047476     6 [GAATTC]
[3] 5056207 5056212     6 [GAATTC]
[4] 5056677 5056682     6 [GAATTC]
[5] 5068417 5068422     6 [GAATTC]
[6] 5075296 5075301     6 [GAATTC]
```

matchPDict

```
> m1 <- DNAStringSet("GAATTC")
> m2 <- DNAStringSet("GGATCC")
> dict <- PDict(append(m1, m2))
> restrict <- matchPDict(dict, eco)
> restrict

MIndex object of length 2

> tail(restrict[[1]])

IRanges instance:
      start     end width
[1] 5012393 5012398     6
[2] 5047471 5047476     6
[3] 5056207 5056212     6
[4] 5056677 5056682     6
[5] 5068417 5068422     6
[6] 5075296 5075301     6
```

Similarly, can be done for vmatchPattern and vmatchPDict.

Analysis of Molecular Variance

Analysis of Molecular Variance (AMOVA) is a method for studying molecular variation within a species. The analysis of molecular variance (AMOVA) was used to study the patterns and degree of relatedness revealed by Multidimensional scaling and the Clustering dendrogram. Further it is used to summarize the population structure with the marker data from different genotypes, while remaining flexible enough to accommodate different types of assumptions about the evolution of the genetic system. Electrophoresis, one of the most widely used methods for studying the structure of DNA, produces marker data in the form of 0s and 1s where 1 denotes the presence of a band and zero its absence. The vector of such 0's and 1's is called DNA haplotype of the individual/variety. Recently, Analysis of Molecular Variance (AMOVA) is used to calculate the 'between groups' and 'within groups' variance. This technique treats genetic distances as deviations from a group mean position, and uses the squared deviations as variances. The total sums of squares of genetic distances can then be partitioned into components that represent the 'within group' and the 'between-group' sum of squares. The resulting test statistic Φ_{ST} is analogous to Wright's F_{ST}, and is the ratio of the between-group mean square to the total mean square (Wright, 1951; Cockerham, 1973). Φ_{ST} represents the correlation between random genetic accessions within a group relative to random accessions from the population at large. This statistic can take values between 0 and 1; higher values indicate greater partitioning of the population into sub-groups.

The Analysis of Molecular Variance Procedure

When a population is divided into isolated subpopulations, there is less heterozygosity than there would be if the population was undivided. Founder effects acting on different demes generally lead to subpopulations with allele frequencies that are different from the larger population. Also, these demes are smaller in size than the larger population; since allele frequency in each generation represents a sample of the previous generation's allele frequency, there will be greater sampling error in these small groups than there would be in a larger undifferentiated population. Hence, genetic drift will push these smaller demes toward different allele frequencies and allele fixation more quickly than would take place in a larger undifferentiated population. For a given species, when several subpopulations are separated geographically, in absence of selection and with random mating, two trends are expected: (i) Gene frequencies for the total population remain constant over generations, and (ii) The variance of gene frequencies increase over time because of differentiation among subpopulations. Wright's F statistic (Wright, 1965), quantify the differentiation among subpopulations and among individuals. However, molecular data reveals not only the frequency of molecular markers, but can also tell us something about the amount of

mutational differences between different genes. Analysis of Molecular Variance (AMOVA) is a method of estimating population differentiation directly from molecular data and testing hypotheses about such differentiation. AMOVA may be used to analyze STMS or AFLP molecular data.

AMOVA treats any kind of raw molecular data as a Boolean vector p_i, that is, a 1 × n matrix of 1's and 0's, 1 indicating the presence of a '*i*' marker and 0 its absence. A marker could be a nucleotide base, a base sequence, a restriction fragment, or a mutational event. Euclidean distances between pairs of vectors are then calculated by subtracting the Boolean vector of one haplotype from another, according to the formula ($p_j - p_k$). If p_j and p_k are visualized as points in *n*-dimensional space indicated by the intersections of the values in each vector, with *n* being equal to the length of the vector, then the Euclidean distance is simply a scalar that is equal to the shortest distance between those two points. The squared Euclidean distances are then calculated using the equation $\delta_{jk}^2 = (p_j - p_k)'W(p_j - p_k)$, where *W* is a weighting matrix; by default, it is an identity matrix and does not change the value of the final product; however, *W* can be a matrix with a number of values depending upon how one weights molecular change at different locations on a sequence or phylogenetic tree.

Partitioning a Distance Matrix into Hierarchical Components

Consider a haploid genetic system where inter-haplotypic distances are identical to distances between individuals. One can arrange a set of N individuals from I populations into a distance matrix, D^2, partitioned into a series of submatrices corresponding to particular subdivisions as below:

$$D^2 = \begin{bmatrix} \begin{bmatrix} D_{11}^2 \end{bmatrix} & \begin{bmatrix} D_{12}^2 \end{bmatrix} & \cdots & \begin{bmatrix} D_{1I}^2 \end{bmatrix} \\ \begin{bmatrix} D_{21}^2 \end{bmatrix} & \begin{bmatrix} D_{22}^2 \end{bmatrix} & & \begin{bmatrix} D_{21}^2 \end{bmatrix} \\ \cdots & & & \cdots \\ \begin{bmatrix} D_{I1}^2 \end{bmatrix} & \cdots & \cdots & \begin{bmatrix} D_{II}^2 \end{bmatrix} \end{bmatrix}$$

where the elements of the block-diagonal submatrices D_{ii}^2 contain pairwise squared-distances (δ_{jk}^2) between individuals of the same (*i*th) population, and those of the off-diagonal matrix blocks $D_{ii'}^2$, contain pairwise squared-distances between individuals, one from the *i*th and other from the *i*'th population. Individuals may also be grouped at higher levels, according to such non-genetic criteria as geography, ecological environment, or language.

A conventional sum of squares [$SS_{(Total)}$] may be written, barring a constant (2N), as the sum of squared differences between all pairs of N items. In the multidimensional case, using

vectors instead of scalars, the conventional sum of squares becomes a sum of squared deviations (SSD) from the centroid of a multidimensional space. Thus,

$$\text{SSD}_{(\text{Total})} = \frac{1}{2N}\sum_{j=1}^{N}\sum_{k=1}^{N}(p_j - p_k)'W(p_j - p_k)$$

$$= \frac{1}{2N}\sum_{j=1}^{N}\sum_{k=1}^{N}\delta_{jk}^2$$

because $\delta_{jj}^2 = 0$ for all haplotype h_j. This transformation applies equally to the total array of individuals in the data set, to those within each population separately (within the diagonal blocks, D_{ii}^2), and to those belonging to a particular subdivision (within the diagonal blocks, $D_{11}^2, D_{12}^2, D_{21}^2$ and D_{22}^2).

Model for AMOVA

Where individuals are arranged into populations and populations nested within groups defined *a priori* on nongenetic criteria, a linear model can be defined on the pattern first described by Cockerham (1969, 1973) and refined upon by others (Weir and Cockerham 1984; Long 1986)

$$p_{jig} = p + a_g + b_{ig} + c_{jig} \quad (1)$$

where p_{jig} indexes the jth chromosome, here equivalent to the jth individual ($j = 1, \ldots, N_{ig}$) in the ith population ($i = 1, \ldots I_g$) in the gth group ($g = 1, \ldots G$) and p is the unknown expectation of p_{jig} averaged over the whole study. The effects are a for group, b for populations and c for individuals within populations. The effects will be assumed to be additive, random, uncorrelated, and to have the associated variance components σ^2_a, σ^2_b and σ^2_c respectively.

Table 1 General design for hierarchical analysis of molecular variance (AMOVA)

Source of variation	**d.f.**	**MSD**	**Expected MSD**
Among regions	G-1	MSD/(AG)	$\sigma_c^2 + n'\sigma_b^2 + n''\sigma_a^2$
Among populations within regions	$\sum_{g=1}^{G} I_g - G$	MSD/(AP/WG)	$\sigma_c^2 + n\sigma_b^2$
Among individuals within populations	$N - \sum_{g=1}^{G} I_g$	MSD(WP)	σ_c^2
Total	N-1		

Relying on the standard decomposition, one can write note that for any choice of hierarchical partition of the *N* individuals into strata,

SSD(Total) = SSD(Among Strata) + SSD(Within Strata),

placing in traditional analysis of variance framework, designated here as Analysis of Molecular variance, AMOVA (Table 1). The total sum of squared deviations, SSD(Total), can be partitioned into components for variation within populations, SSD(WP), variation among populations within regional groups, SSD(AP/WG), and variation among regional groups, SSD(AG). The corresponding sums of squares are

$$\text{SSD(WP)} = \sum_{g=1}^{G}\sum_{i=1}^{I_g}\frac{\sum_{j=1}^{N_{ig}}\sum_{k=1}^{N_{ig}}\delta_{jk}^2}{2N_{ig}}$$

$$\text{SSD(AP/WG)} = \sum_{g=1}^{G}\left(\frac{\sum_{i=1}^{I_g}\sum_{j=1}^{N_{ig}}\sum_{i'=1}^{I_g}\sum_{k=1}^{N_{i'g}}\delta_{jk}^2}{\sum_{i=1}^{I_g}2N_{ig}} - \sum_{i=1}^{I_g}\frac{\sum_{j=1}^{N_{ig}}\sum_{k=1}^{N_{ig}}\delta_{jk}^2}{2N_{ig}}\right)$$

and

$$\text{SSD(AG)} = \left(\frac{\sum_{j=1}^{N_{ig}}\sum_{k=1}^{N_{ig}}\delta_{jk}^2}{2N_{ig}} - \sum_{g=1}^{G}\frac{\sum_{i=1}^{I_g}\sum_{j=1}^{N_{ig}}\sum_{i'=1}^{I_g}\sum_{k=1}^{N_{i'g}}\delta_{jk}^2}{\sum_{i=1}^{I_g}2N_{ig}}\right).$$

The mean squared deviations (MSD) are then obtained by dividing such SSD by the appropriate degrees of freedom as reported in Table 1. The *n* coefficients in Table 1 represent the average sample sizes of particular hierarchical levels, allowing for unequal sample sizes,

$$n = \frac{\sum_{g=1}^{G}\sum_{i=1}^{I_g}N_{ig} - \sum_{g=1}^{G}\left(\frac{\sum_{i=1}^{I_g}N_{ig}^2}{\sum_{i=1}^{I_g}N_{ig}}\right)}{\sum_{g=1}^{G}I_g}$$

$$n' = \frac{\sum_{g=1}^{G}\left(\frac{\sum_{j=1}^{I_g} N_{ig}^2}{\sum_{i=1}^{I_g} N_{ig}}\right) - \frac{\sum_{g=1}^{G}\sum_{j=1}^{I_g} N_{ig}^2}{\sum_{g=1}^{G}\sum_{i=1}^{I_g} N_{ig}}}{G-1}$$

$$n'' = \frac{\sum_{g=1}^{G}\sum_{i=1}^{I_g} N_{ig} - \frac{\sum_{g=1}^{G}\left(\sum_{j=1}^{I_g} N_{ig}\right)^2}{\sum_{g=1}^{G}\sum_{i=1}^{I_g} N_{ig}}}{G-1}$$

The variance components (σ^2's) of each hierarchical level are extracted by equating the mean squares (MSDs) to their expectations. It may also be useful to employ haplotypic correlation measures, which are termed as Φ-statistics. The different variance components can be expressed in terms of Φ-statistics as

$$\sigma_c^2 = (1-\Phi_{SC})\sigma^2$$

$$\sigma_b^2 = (\Phi_{ST}-\Phi_{CT})\sigma^2$$

$$\sigma_a^2 = \Phi_{CT}\sigma^2$$

where $\sigma^2 = \sigma_a^2 + \sigma_b^2 + \sigma_c^2$; Φ_{ST} is viewed as the correlation of random haplotypes within populations, relative to that of random pairs of haplotypes drawn from the whole species; Φ_{CT} as the correlation of random haplotypes within a group of populations, relative to that of random pairs of haplotypes drawn from the whole species, and Φ_{SC} as the correlation of the molecular diversity of random haplotypes within populations, relative to that of random pairs of haplotypes drawn from the region. One can rewrite the above equations in terms of the Φ-statistics as

$$\Phi_{ST} = \frac{\sigma_a^2 + \sigma_b^2}{\sigma^2},\ \Phi_{CT} = \frac{\sigma_a^2}{\sigma^2},\ \Phi_{SC} = \frac{\sigma_b^2}{\sigma_b^2 + \sigma_c^2}$$

Testing Significance of the Variance Components and Φ- Statistics

For testing variance components, the traditional analysis of molecular variance procedure cannot be adopted because the molecular data consist of Euclidean distances derived from vectors of 1's and 0's, and the data re unlikely to follow a normal distribution. A null distribution is therefore computed by resampling of the data or by a permutation procedure. Excoffier, *et al*., (1992) has discussed the methods for testing the significance of the variance components obtained from analysis of molecular variance. Under this procedure each

individual is allocated to a randomly chosen population, while holding sample sizes constant at the realized values so as to obtain null distribution. This amounts to random permutation of the rows (and corresponding columns) of the squared distance matrix. The variance-components are estimated from each of a large number (say 500) of permuted matrices.

Further, they suggested two other permutation schemes that are useful for testing Φ_{SC}, σ_b^2 and Φ_{CT}, σ_c^2. The first assumes that the regions are real but that the populations within them are not, permuting individuals within regional groups without regard to population. The second assumes that while the populations are real, the regional groupings are artificial, permuting whole populations across groups. In this case, the sizes of the groups (but not those of the populations) vary with each permutational run.

Restriction Site Sampling

The sampling of nucleotides shows a major source of variability for the estimation of molecular diversity (Lynch and Crease 1990). One can ask whether the results depend on a particular array of marker sites employed. Excoffier, *et al.*, (1992) examined the influence of site sampling on the genetic structure of the populations, using a site resampling plain similar to the bootstrap used by Efron (1982). Under the assumption the observed n sites are representative of all molecular markers. They obtain the distribution of the variance components and associated Φ- statistics by Monte Carlo simulation, using 500 random collection sites. For each collection, the procedure is as follows: (a) Draw a given number of sites from the observed array of m sites, at random and with replacement. Given the choice of sites, the haplotype of each individual is then taken as the combination of the original states of those randomly chosen sites; (b) compute interhaplotypic distances on the basis of the newly defined haplotypes and perform an AMOVA analysis. The distances are simply computed from euclidean distance; and (c) permute the matrix 500 times, and test the significance of the different statistics with the previously described procedures.

References

Cockerham, C.C. 1969. Variance of gene frequencies. *Evolution*, 23: 72–84.

Cockerham, C.C. 1973. Analysis of gene frequencies. *Genetics*, 74: 679–700

Efron, B. 1982. The Jacknife, the Bootstrap and Other Resampling Plans. Regional conference Series in Applied Mathematics, Vol. 38. Society for Industrial and Applied Mathematics, Philadelphia.

Excoffier, L., Smouse, P. and Quattro, J. 1992. Analysis of molecular variance inferred for metric distances among DNA haplotypes: application to human mitochondrial DNA restriction data. *Genetics*, 131:479–491.

Wright, S. 1951. The genetical structure of populations. *Ann. Eugen.*, 15:323–354.

Genome Assembly

Introduction

Genome assembly refers to aligning and merging fragments of genomic DNA sequence in order to construct the original sequence. This is mandatory as DNA sequencing technology cannot read whole genomes in one go, but rather reads small pieces, in random order, of between 20 and 1000 bases, depending on the technology used. Recent advances in sequencing technology made it possible to generate vast amounts of sequence data. The fragments produced by these high-throughput methods are, however, far shorter than in traditional Sanger sequencing.

The first sequence assemblers began to appear in the late 1980s and early 1990s as variants of simpler sequence alignment programs to piece together vast quantities of fragments generated by automated sequencing instruments called DNA sequencers. Algorithms were developed for whole genome shotgun (WGS) fragment assembly, including Atlas, Arachne, Celera, PCAP, Phrap (www.phrap.org) and Phusion. All these programs rely on the overlap-layout-consensus approach where all the reads are compared to each other in a pair-wise fashion.

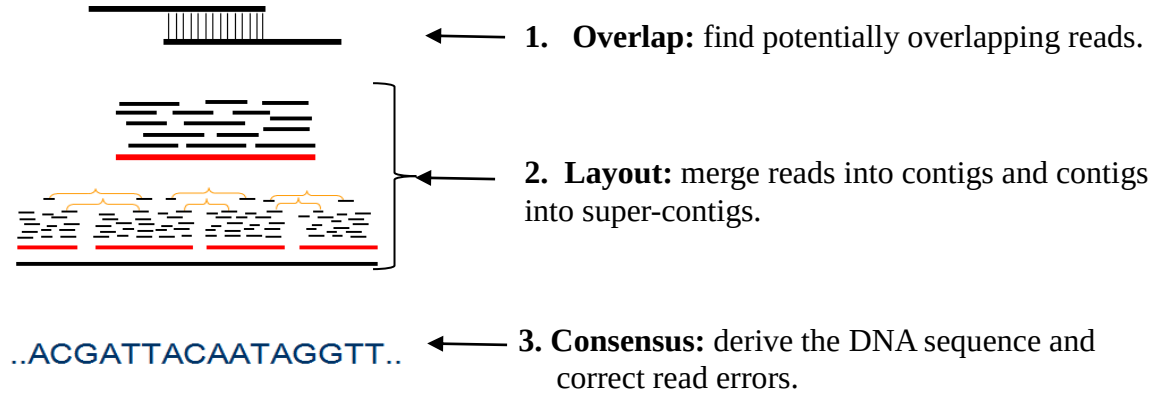


Figure 1: Scaffolds

The resulting (draft) genome sequence is produced by combining the information sequenced **"contigs"** and then employing linking information to create **"scaffolds"** (Figure 1.). Scaffolds are positioned along the physical map of the chromosomes creating a "golden path".

Recently, new sequencing methods have emerged. Commercially available technologies include pyrosequencing (454 Sequencing), sequencing by synthesis (Illumina) and

sequencing by ligation (SOLiD). The reads produced by these next-generation sequencing technologies are much shorter than traditional Sanger reads. Because of their shorter length, they must be produced in large quantities and at greater coverage depths than the earlier sequencing techniques. Whereas long reads provide long overlaps, to disambiguate repeats from real overlaps, short reads within repeats offer fewer differences to judge from. These issues have led several research teams to design de novo assembly tools specifically for these very short reads.

Types of Sequencers and Data Format

Illumina	:	FASTQ
SoLID/ABI-Life	:	FASTA
Roche 454	:	SFF
Ion Torrent	:	SFF or FASTQ

Types of Assembly

There are two type of assembly base on the availability of reference genome:

a) **De novo Assembly:** Reads are aligned to each other to form a consensus sequences that are called contigs.

b) **Reference genome assembly:** Here reads are aligned with the available reference genome to form a consensus sequences.

Genome Assembly Techniques

Almost all large-scale sequencing projects employ the shotgun strategy that assembler (deduce) the target DNA sequence from a set of short DNA fragments determined from DNA pieces randomly sampled from the target sequence. The set of short DNA fragments, called shotgun reads, are assembled into a set of contigs, or set of aligned fragments, using a computer program, fragment assembler. The fragment assembly is a conceptually simple procedure that generates longer sequences by detecting overlapping fragments. If the fragment assembly can be done perfectly, the genome sequencing would be a simple problem. However, there are extensive repetitive sequences, repeats in short, in a genomic sequence, which can easily mislead the fragment assembly process. A useful technique to overcome the difficulty from repeats is to sequence both ends of a clone, generating two fragment reads per clone. Since the insert size of clone is known, we know the approximate distance between two fragments. The fragment matching information is often referred as mate-pair information, which becomes essential for large-scale shotgun sequencing. The main issue utilizing this information during the assembly process is that we do not know the sequence between two reads, which can only be deduced by assembling other fragments into

single contigs. So we can utilize the clone-length information only after assembly, which result either a correct or an incorrect assembly based on the clone-length information. One strategy to use the mate-pair information effectively is to assemble contigs as accurate as possible by detecting potentially misassembled contigs and then utilize the mate-pair information using only contigs that are likely to be assembled correctly. General procedure for genome sequencing and assembly emphasizing the procedures that used at genome-sequencing centres-

1. **Fragment readout:** The sequences of each fragment are determined using automatic base-calling software. Phred is the most widely used program.
2. **Trimming vector sequences:** Shotgun reads often contain part of the vector sequences that have to be removed before sequence assembly.
3. **Trimming low-quality sequences:** Shotgun reads contain poor quality base calls and removing or masking out these low-quality base calls often leads to more accurate sequence assembly. However, this step is optional and some sequencing centres do not mask out low-quality base calls, relying on the fragment assembler to utilize quality values to decide true fragment overlaps.
4. **Fragment assembly:** The shotgun data is input to a fragment assembler that automatically generates a set of aligned fragment called contigs.
5. **Assembly validation:** Some contigs that assembled in the previous steps may be misassembled due to repeats. Since we do not have a priori knowledge on repeats in the targets DNA, it is very difficult to verify the correctness of assembly of each contig and this step is largely done manually. There are recent algorithmic developments on automatic verification of contig assemblies.
6. **Scaffolding Contigs:** Contigs needs to be oriented and ordered. The mate-pair information is a primary information source for this step, thus this step is not achievable if the input shotgun is not prepared by reading both ends of clones.
7. **Finishing:** Assuming that all contigs are assembled correctly and contigs are oriented and ordered correctly, we can close gaps between two contigs by sequencing specific regions that corresponds to the position of gaps.

De novo assembly of next-generation sequencing reads

After NGS reads have been generated, they are aligned to a known reference sequence or assembled *de novo.* De novo assembly is the process of reconstructing the genome of organisms not sequenced before or for which a reference comparative genome is unavailable. It is accomplished through the shotgun process where the genome of the organism is sheared into small fragments, each of which is sequenced separately and reconstructed using computational tools. This process is complex because genomes contain segments of identical sequences namely repeats. The length of the repeats varies very much and makes it impossible to recover the complete genome. Therefore, almost all de novo tools do not recover the complete genome. However, they report long segments of genome known as contigs. Furthermore, the complexity increases with the size of the genome. There are

primarily two categories in de novo genome assembling process namely Overlap Layout and Consensus (OLC) and De Bruijn graph based methods. OLC based methods are computationally more expensive than De Bruijn graph based methods, whereas the latter are memory intensive. There are several tools available based on De Bruijn graph.

Assembly for Double-Ended Short-read Sequencing Technologies

Recently developed Pyrosequencing-like technologies are extremely promising. However, the length of the resulting reads is drastically shorter than those produced by current sequencing machines. Sequence repeats limit the usefulness of reads, as any sequence repetition exceeding the read-length defines an irresolvable ambiguity. In particular, the shortest common superstring of collection of short reads is likely to be a highly over-compressed representation of target. To solve the problem of repeats, the variable-insert length, double-ended read protocol were proposed. Fragment multiple-target clones and use gel electrophoresis to separate out all fragments of length $a \pm b\%$, or (equivalently stated) of length d to $d+w$ for given integers d and w.

De Bruijn Graph

In 1995, Idury and Waterman introduced the use of a graph to represent an assembly. They presented an assembly algorithm for an alternative sequencing technique, sequencing by hybridization, where an oligoarray could detect all the k nucleotide words, also known as k-mers, present in a given genome. Their resolution method consisted in creating a node for every detected word, and then connects the nodes corresponding to overlapping k-mers. They could then report chains of overlapping k-mers which unambiguously produced contigs, because of an absence of branching connections. This sequence graph is called de Bruijn graph, whereby the k-mers are represented as arcs and overlapping k-mers join at their tips. It contains novel algorithms for graph construction, error removal, mixed length assembly and paired-end assembly. Moreover, this program was designed to be robust and easy to run. These has some special aspects Firstly, it maps k-mers onto nodes instead of arcs. Secondly, it associates reverse complementary sequences to obtain an implicit bi-graph (or bi-directed graph), in other words a graph where an edge can independently enter or exit the nodes at either of its ends. Each node N represents a series of overlapping k-mers. Adjacent k-mers overlap by k − 1 nucleotides. The marginal information contained by a k-mer is its last nucleotide. The sequence of those final nucleotides is called the sequence of the node or s(N). The sequence of a node is therefore an incomplete representation of the corresponding k-mers. In other words, two distinct sets of k-mers can be represented by two separate nodes having the same sequence. Despite having the same sequence, these two nodes are nonetheless kept separate, and the reads are mapped onto them according to the underlying k-mers. Each node N is attached to a twin node ˜N, which represents the reverse series of reverse complement k-mers. This ensures that overlaps between reads from opposite strands

are taken into account. It is important to note that the sequences attached to a node and its twin does not need to be reverse complements of each other. The union of a node N and its twin is called a block. From now on any change to a node is implicitly applied symmetrically to its twin. The blocks can be considered as the nodes of an implicit bi-graph. Nodes can be connected by a directed edge or arc. In that case, the last k-mer of an arc's origin node overlaps with the first of its destination node. Because of the symmetry of the blocks, if an arc goes from node A to B, a symmetric arc goes from ˜B to ˜A. Any modification of one arc is implicitly applied symmetrically to its paired arc.

Each node, represented by a single rectangle, represents a series of overlapping k-mers (in this case, k = 5) listed directly above or below. The last nucleotide of each k-mer is colored in red. The sequence of those final nucleotides, copied in large letters in the rectangle, is the sequence of the node. The twin node, directly attached either below or above the node, represents the reverse series of reverse complement k-mers. Arcs are represented as arrows between nodes. The last k-mer of an arc's origin overlaps with the first of its destination. Each arc has a symmetric arc. The two nodes on the left could be merged into one without loss of information, because they form a chain.

- In the de Bruijn graph, there is a one-to-one mapping of sequences onto paths traversing the graph. Extracting the nucleotide sequence from a path is straightforward given the initial k-mer of the first node and the sequences of all the nodes in the path. Conversely, for every read there exists exactly one path which goes sequentially through the nodes corresponding to the sequence's k-mers.
- Two overlapping sequences are represented as two paths which overlap. The intersection of the paths corresponds to the overlap between the sequences. The two paths form a sub graph the topology of which is directly linked to the type of alignment between the sequences. If one sequence is a substring of the other, then its path is a sub-path of the other's path. If the sequences align along their tips, then their paths will also be connected at their extremities. When adding more sequences, all of the above properties remain valid. This means that all sequences which share a common substring are constrained to go through the path corresponding to that substring. This property will be useful when searching for sets of overlapping reads through repeats, as they all follow the same path.

The first consequence is that a de Bruijn graph can accommodate sequences of very different lengths. This is especially useful when attempting mixed-length sequencing or comparative genomics. No ad hoc approximation has to be made to mix short reads, long reads, pre-assembled contigs or even finished genomes. Secondly, because of the one-to-one relationship between paths and sequences, overlapping sequences necessarily follow the same path. This simplifies the search for consistently overlapping sets of reads.

Issues and Problems of Assembling Complex Genomes

Genome assembly is a very difficult computational problem, made more difficult because many genomes contain large numbers of identical sequences, known as repeats. These repeats can be thousands of nucleotides long, and some occur in thousands of different locations, especially in the large genomes of plants and animals.

One challenge to sequencing crop genomes is the vast difference in scale between the size of the genomes and the lengths of the reads produced by the different sequencing methods. While there may be a 10–500× difference in scale between the short reads produced by second-generation sequencing and modern Sanger sequencing, this is still dwarfed by the difference between Sanger read length and the lengths of complete chromosomes. As the sequenced organisms grew in size and complexity the assembly programs used in genome projects needed increasingly sophisticated strategies to handle:

- Terabytes of sequencing data which need processing on computing clusters;
- Identical and nearly identical sequences (known as *repeats*) which can, in the worst case, increase the time and space complexity of algorithms exponentially; and
- Errors in the fragments from the sequencing instruments, which can confound assembly.

Table 1: Lists of prevalent de-novo assemblers

Name	Type	Technologies	Author	Late Updated
BySS	(large) genomes	Solexa, SOLiD	Simpson, J. et al.	2008 / 2011
ALLPATHS-LG	(large) genomes	Solexa, SOLiD	Gnerre, S. et al.	2011
AMOS	genomes	Sanger, 454	Salzberg, S. et al.	2002 / 2008
Arapan-M	Medium Genomes (e.g. E.coli)	All	Sahli, M. & Shibuya, T.	2011 / 2012
Arapan-S	Small Genomes (Viruses and Bacteria)	All	Sahli, M. & Shibuya, T.	2011 / 2012
Celera WGA Assembler / CABOG	(large) genomes	Sanger, 454, Solexa	Myers, G. et al.; Miller G. et al.	2004 / 2010
CLC Genomics Workbench & CLC Assembly Cell	genomes	Sanger, 454, Solexa, SOLiD	CLC bio	2008 / 2010 / 2011
Cortex	genomes	Solexa, SOLiD	Iqbal, Z. et al.	2011
DNA Baser	genomes	Sanger, 454	Heracle BioSoft SRL	2013
DNA Dragon	genomes	Illumina, SOLiD, Complete Genomics,	SequentiX	2011

		454, Sanger		
DNAnexus	genomes	Illumina, SOLiD, Complete Genomics	DNAnexus	2011
Edena	genomes	Illumina	D. Hernandez, P. François, L. Farinelli, M. Osteras, and J. Schrenzel.	2008/ 2013
Euler	genomes	Sanger, 454 (Solexa)	Pevzner, P. et al.	2001 / 2006
Euler-sr	genomes	454, Solexa	Chaisson, MJ. et al.	2008
Forge	(large) genomes, EST, metagenomes	454, Solexa, SOLID, Sanger	Platt, DM, Evers, D.	2010

References

Batzoglou, S., Jaffe, D.B., Stanley, K, Butler, J, Gnerre, S, Mauceli, E, Berger, B, Mesirov, J.P. et al. (January 2002). "ARACHNE: a whole-genome shotgun assembler". Genome Research 12 (1): 177–89. doi:10.1101/gr.208902. PMC 155255. PMID 11779843.

Boisvert, Sébastien, Laviolette, François, Corbeil, Jacques (2010). "Ray: simultaneous assembly of reads from a mix of high-throughput sequencing technologies". Journal of Computational Biology 17 (11): 1519–33. doi:10.1089/cmb.2009.0238. PMC 3119603. PMID 20958248.

Dohm, J. C., Lottaz, C.; Borodina, T., Himmelbauer, H. (November 2007). "SHARCGS, a fast and highly accurate short-read assembly algorithm for de novo genomic sequencing". Genome Research 17 (11): 1697–706. doi:10.1101/gr.6435207. PMC 2045152. PMID 17908823.

Huse, S. M., Huber, J. A., Morrison, H. G., Sogin, M. L., and Welch, D. M. (2007). Accuracy and quality of massively parallel DNA pyrosequencing, Genome Biol 8, R143.

Mardis, E. R. (2008). The impact of nextgeneration sequencing technology on genetics, Trends Genet 24, 133–141.

Michael C. Schatz, Jan Witkowski and W Richard McCombie (2012). Current challenges in de novo plant genome sequencing and assembly. Genome Biology, 13:243

Myers, E. W., Sutton, G.G., Delcher, A.L., Dew, I.M., Fasulo, D.P., Flanigan, M.J., Kravitz, S.A. Mobarry, C.M. et al. (March 2000). "A whole-genome assembly of Drosophila". Science 287 (5461): 2196–204. doi:10.1126/science.287.5461.2196. PMID 10731133.

Pop, M. (2004) Shotgun sequence assembly, Adv Comput 60, 193–248. 7.

Pop, M. and Salzberg, S. L. (2008). Bioinformatics challenges of new sequencing technology, Trends Genet 24, 142–149.

Ronaghi, M., Uhlen, M. and Nyren, P. (1998). A sequencing method based on real-time pyrophosphate, Science 281, 363–365.

Zhang W., Chen J., Yang Y., Tang Y., Shang J., et al. (2011). A Practical Comparison of De Novo Genome Assembly Software Tools for Next-Generation Sequencing Technologies. PLoS ONE 6(3): e17915. doi:10.1371/journal.pone.0017915.

Genome Annotation

Introduction

Until the genome revolution, genes were identified by researchers with specific interests in a particular protein or cellular process. Once identified, these genes were isolated, typically by cloning and sequencing cDNAs, usually followed by targeted sequencing of the longer genomics segments that code for the cDNAs. Once an organism's entire genome sequence becomes available, there is strong motivation for finding all the genes encoded by a genome at once rather than in a piecemeal approach. Such catalogue is immensely valuable to researchers, as they can learn much more from the whole picture than from a much more limited set of genes. For example, genes of similar sequence can be identified, evolutionary and functional relationships can be elucidated, and a global picture of how many and what types of genes are present in a genome can be seen. A significant portion of the effort in genome sequencing is devoted to the process of *annotation*, in which genes, regulatory elements, and other features of the sequence are identifies as thoroughly as possible and catalogued in a standard format in public databases so that researchers can easily use the information. Functional genomics research has expanded enormously in the last decade and particularly the plant biology research community. Functional annotation of novel DNA sequences is probably one of the top requirements in functional genomics as this holds, to a great extent, the key to the biological interpretation of experimental results.

Computational Gene Prediction

Computational gene prediction is becoming more and more essential for the automatic analysis and annotation of large uncharacterized genomic sequences. In the past two decades, many algorithms have been evolved to predict protein coding regions of the DNA sequences. They all have in common, to varying degree, the ability to differentiate between gene features like Exons, Introns, Splicing sites, Regulatory sites etc. Gene prediction methods predicts coding region in the query sequences and then annotates the sequences databases.

Gene Structure and Expression

The gene structure and the gene expression mechanism in eukaryotes are far more complicated than in prokaryotes. In typical eukaryotes, the region of the DNA coding for a protein is usually not continuous. This region is composed of alternating stretches of *exons* and *introns*. During transcription, both exons and introns are transcribed onto the RNA, in their linear order. Thereafter, a process called *splicing* takes place, in which, the intron sequences are excised and discarded from the RNA sequence. The remaining RNA segments, the ones corresponding to the exons are ligated to form the mature RNA strand. A typical multi-exon gene has the following structure. It starts with the promoter region, which is

followed by a transcribed but non-coding region called *5′ untranslated region (5′ UTR)*. Then follows the initial exon which contains the start codon. Following the initial exon, there is an alternating series of introns and internal exons, followed by the terminating exon, which contains the stop codon. It is followed by another non-coding region called the *3′ UTR*. Ending the eukaryotic gene, there is a polyadenylation (polyA) signal: the nucleotide Adenine repeating several times. The exon-intron boundaries (i.e., the splice sites) are signalled by specific short (2bp long) sequences. The 5'(3') end of an intron (exon) is called the *donor* site, and the 3'(5') end of an intron (exon) is called the *acceptor* site. The problem of gene identification is complicated in the case of eukaryotes by the vast variation that is found in gene structure.

Gene Prediction Methods

There are mainly two classes of methods for computational gene prediction. One is based on sequence similarity searches, while the other is gene structure and signal-based searches, which is also referred to as Ab initio gene finding.

Sequence Similarity Searches

Sequence similarity search is a conceptually simple approach that is based on finding similarity in gene sequences between ESTs (expressed sequence tags), proteins, or other genomes to the input genome. This approach is based on the assumption that functional regions (exons) are more conserved evolutionarily than non-functional regions (intergenic or intronic regions). Once there is similarity between a certain genomic region and an EST, DNA, or protein, the similarity information can be used to infer gene structure or function of that region. EST-based sequence similarity usually has drawbacks in that ESTs only correspond to small portions of the gene sequence, which means that it is often difficult to predict the complete gene structure of a given region. Local alignment and global alignment are two methods based on similarity searches. The most common local alignment tool is the BLAST family of programs, which detects sequence similarity to known genes, proteins, or ESTs. The biggest limitation to this type of approaches is that only about half of the genes being discovered have significant homology to genes in the databases.

Ab initio Gene Prediction Methods

The second class of methods for the computational identification of genes is to use gene structure as a template to detect genes, which is also called *ab initio* prediction. *Ab initio* gene predictions rely on two types of sequence information: signal sensors and content sensors. Signal sensors refer to short sequence motifs, such as splice sites, branch points, poly pyrimidine tracts, start codons and stop codons. Exon detection must rely on the content sensors, which refer to the patterns of codon usage that are unique to a species, and allow

coding sequences to be distinguished from the surrounding non-coding sequences by statistical detection algorithms.

Many algorithms are applied for modeling gene structure, such as Dynamic Programming, linear discriminant analysis, Linguist methods, Hidden Markov Model and Neural Network. Based on these models, a great number of *ab initio* gene prediction programs have been developed.

Gene Discovery in Prokaryotic Genomes

Discovery of genes in Prokaryote is relatively easy, due to the higher gene density typical of prokaryotes and the absence of introns in their protein coding regions. DNA sequences that encode proteins are transcribed into mRNA, and the mRNA is usually translated into proteins without significant modification. The longest ORFs (open reading frames) running from the first available start codon on the mRNA to the next stop codon in the same reading frame generally provide a good, but not assured prediction of the protein coding regions. Several methods have been devised that use different types of Markov models in order to capture the compositional differences among coding regions, “shadow" coding regions (coding on the opposite DNA strand), and noncoding DNA. Such methods, including ECOPARSE, the widely used GENMARK, and Glimmer program, appear to be able to identify most protein coding genes with good performance.

Gene Discovery in Eukaryotic Genome

It is a quite different problem from that encountered in prokaryotes. Transcription of protein coding regions initiated at specific promoter sequences is followed by removal of noncoding sequences (introns) from pre-mRNA by a splicing mechanism, leaving the protein encoding exons. Once the introns have been removed and certain other modifications to the mature RNA have been made, the resulting mature mRNA can be translated in the 5` to 3` direction, usually from the first start codon to the first stop codon. As a result of the presence of intron sequences in the genomic DNA sequences of eukaryotes, the ORF corresponding to an encoded gene will be interrupted by the presence of introns that usually generate stop codons.

Gene Prediction Program

There are two basic problems in gene prediction: prediction of protein coding regions and prediction of the functional sites of genes. Gene prediction program can be classified into four generations. The first generation of programs was designed to identify approximate locations of coding regions in genomic DNA. The most widely known programs were probably TestCode and GRAIL. But they could not accurately predict precise exon locations. The second generation, such as SORFIND and Xpound, combined splice signal and coding region identification to predict potential exons, but did not attempt to assemble predicted

exons into complete genes. The next generation of programs attempted the more difficult task of predicting complete gene structures. A variety of programs have been developed, including GeneID, GeneParser, GenLang, and FGENEH. However, the performance of those programs remained rather poor. Moreover, those programs were all based on the assumption that the input sequence contains exactly one complete gene, which is not often the case. To solve this problem and improve accuracy and applicability further, GENSCAN and AUGUSTUS were developed, which could be classified into the fourth generation.

GeneMark

GeneMark uses a Markov Chain model to represent the statistics of the coding and noncoding frames. The method uses the dicodon statistics to identify coding regions. Consider the analysis of a sequence x whose base at the ith position is called x_i. The Markov chains used are fifth order, and consist of a terms such as $P(a/x_1x_2x_3x_4x_5)$, which represent the probability of the sixth base of the sequence x being given a given that the previous five bases in the sequence x where $x_1x_2x_3x_4x_5$, resulting in the first dicodon of the sequence being $x_1x_2x_3x_4x_5a$. These terms must be defined for all possible pentamers with the general sequence $b_1b_2b_3b_4b_5$. The values of these terms can be obtained of analysis of data, consisting of nucleotide sequence in which the coding regions have been actually identified. When there are sufficient data, they are given by

$$P(\frac{a}{b_1b_2b_3b_4b_5}) = \frac{n_{b_{1b_2b_3}b_4b_5a}}{\sum_{a=A,C,G,T} n_{b_{1b_2b_3}b_4b_5a}}$$

where, $n_{b_{1b_2b_3}b_4b_5a}$ is the number of times the sequence $b_1b_2b_3b_4b_5a$ occurs in the training data. This is the maximum likelihood estimators of the probability from the training data.

Glimmer

The core of Glimmer is Interpolated Markov Model (IMM), which can be described as a generalized Markov chain with variable order. After GeneMark introduces the fixed-order Markov chains, Glimmer attempts to find a better approach for modeling the genome content. The motivational fact is that the bigger the order of the Markov chain, the more non-randomness can be described. However, as we move to higher order models, the number of probabilities that we must estimate from the data increases exponentially. The major limitation of the fixed-order Markov chain is that models from higher order require exponentially more training data, which are limited and usually not available for new sequences. However, there are some oligomers from higher order that occur often enough to be extremely useful predictors. For the purpose of using these higher-order statistics, whenever sufficient data is available, Glimmer IMMs.

Glimmer calculates the probabilities for all Markov chains from 0^{th} order to 8^{th}. If there are longer sequences (e.g. 8-mers) occurring frequently, IMM makes use of them even when there is insufficient data to train an 8-th order model. Similarly, when the statistics from the 8-th order model do not provide significant information, Glimmer refers to the lower-order models to predict genes.

Opposed to the supervised GeneMark, Glimmer uses the input sequence for training. The ORFs longer than a certain threshold are detected and used for training, because there is high probability that they are genes in prokaryotes. Another training option is to use the sequences with homology to known genes from other organisms, available in public databases. Moreover, the user can decide whether to use long ORFs for training purposes or choose any set of genes to train and build the IMM.

GeneMark.hmm

GeneMark.hmm is designed to improve GeneMark in finding exact gene starts. Therefore, the properties of GeneMark.hmm are complementary to GeneMark. GeneMark.hmm uses GeneMark models of coding and non-coding regions and incorporates them into hidden Markov model framework. In short terms, Hidden Markov Models (HMM) are used to describe the transitions from non-coding to coding regions and vice versa. GeneMark.hmm predicts the most likely structure of the genome using the Viterbi algorithm, a dynamic programming algorithm for finding the most likely sequence of hidden states. To further improve the prediction of translation start position, GeneMark.hmm derives a model of the ribosome binding site (6-7 nucleotides preceding the start codon, which are bound by the ribosome when initiating protein translation). This model is used for refinement of the results.

Both GeneMark and GeneMark.hmm detect prokaryotic genes in terms of identifying open reading frames that contain real genes. Moreover, they both use pre-computed species-specific gene models as training data, in order to determine the parameters of the protein-coding and non-coding regions.

Orpheus

The ORPHEUS program uses homology, codon statistics and ribosome binding sites to improve the methods presented so far by using information that those programs ignored. One of the key differences is that it uses database searches to help determine putative genes, and is thus an extrinsic method. This initial set of genes is used to define the coding statistics for the organism, in this case working at the level of codon, not dicodons. These statistics are then used to define a larger set of candidate ORFs. From this set, those ORFs with an unambiguous start codon end are used to define a scoring matrix for the ribosome-binding site, which is then used to determine the 5` end of those ORFs where alternative start are present.

EcoParse

EcoParse is one of the first HMM model based gene finder, was developed for gene finding in *E.coli*. It focuses on the uses the codon structure of genes. With EcoParse a flora of HMM based gene finder, usuing dynamic programming and the viterbi algorithm to parse a sequence, emerged.

Evaluation of Gene Prediction Programs

In the field of gene prediction accuracy can be measured at three levels

a. Coding nucleotides (base level)
b. Exon structure (exon level)
c. Protein product (protein level)

At base level gene predictions can be evaluated in terms of *true positives (TP)* (predicted features that are real), *true negatives* (TN) (non-predicted features that are not real), *false positives (FP)* (predicted features that are not real), and *false negatives (FN)* (real features that were not predicted) Fig. 1. Usually the base assignment is to be in a coding or non coding segment, but this analysis can be extended to include non coding parts of genes, or any functional parts of the sequences.

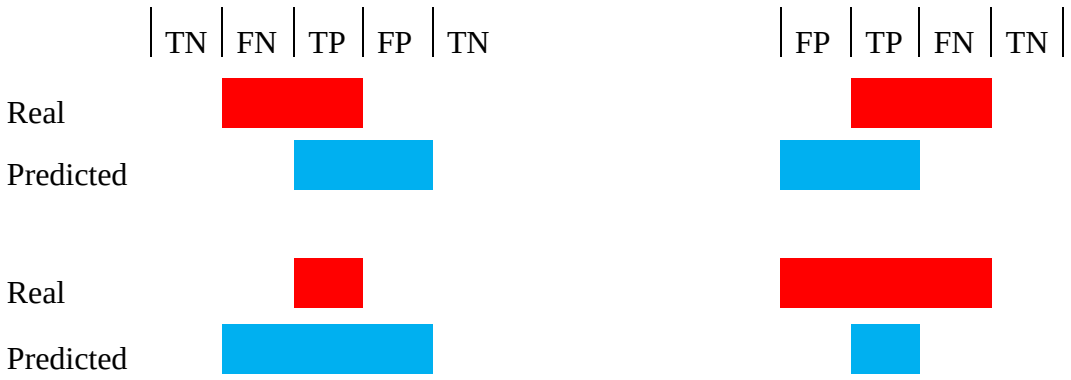


Figure 1: Four Possible Comparisons of Real and Predicted Genes

Sensitivity (Sn): The fraction of bases in real genes that are correctly predicted to be in genes is the sensitivity and interpreted as the probability of correctly predicting a nucleotide to be in a given gene that it actually is.

$$Sn = \frac{TP}{TP + FN}$$

Specificity (Sp): The fraction of those bases which are predicted to be in genes that actually are is called the specificity and interpreted as the probability of a nucleotide actually being in a gene given that it has been predicted to be.

$$Sp = \frac{TP}{TP + FP}$$

Care has to be taken in using these two values to assess a gene prediction program because, as with the normal definition of specificity, extreme results can make them misleading.

Approximate correlation coefficient (AC) has been proposed as a single measure to circumvent these difficulties. This defined as AC=2(ACP-0.5), where

$$ACP = \frac{1}{n}\left(\frac{TP}{TP+FN}+\frac{TP}{TP+FP}+\frac{TN}{TN+FP}+\frac{TN}{TN+FN}\right),$$

At the exon level, determination of prediction accuracy depends on the exact prediction of exon start and end points. There are two measures of sensitivity and specificity used in the field, each of which measures a different but useful property.

The sensitivity measures used are

S_{n1} = CE/AE and Sn2 = ME/AE

The specificity measures used are

S_{p1}=CE/PE and S_{p2}=WE/PE

Where,

AE = No of actual exons in the data

PE = No of predicted exons in the data

CE = No of correct predicted exons

ME = No of missing exons (rarely occurs)

WE = No of wrongly predicted exons (Figure-5)

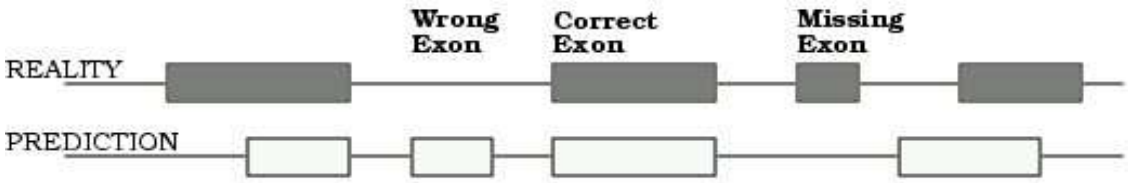


Figure 2: Real and Predicted Exons

Gene Ontology

The gene ontology (GO, http:www.geneontology.org) is probably the most extensive scheme today for the description of gene product functions but other systems such as enzyme codes, KEGG pathways, FunCat, or COG are also widely used. Here, we describe the Blast2GO (B2G, www.blast2go.org) application for the functional annotation, management, and data mining of novel sequence data through the use of common controlled vocabulary schemas.

The main application domain of the tool is the functional genomics of nonmodel organisms and it is primarily intended to support research in experimental labs. Blast2GO strives to be the application of choice for the annotation of novel sequences in functional genomics projects where thousands of fragments need to be characterized. Functional annotation in Blast2GO is based on homology transfer. Within this framework, the actual annotation procedure is configurable and permits the design of different annotation strategies. Blast2GO annotation parameters include the choice of search database, the strength and number of blast results, the extension of the query-hit match, the quality of the transferred annotations, and the inclusion of motif annotation. Vocabularies supported by B2G are gene ontology terms, enzyme codes (EC), InterPro IDs, and KEGG pathways.

Figure 7 shows the basic components of the Blast2GO suite. Functional assignments proceed through an elaborate annotation procedure that comprises a central strategy plus refinement functions. Next, visualization and data mining engines permit exploiting the annotation results to gain functional knowledge. GO annotations are generated through a 3-step process: blast, mapping, annotation. InterPro terms are obtained from InterProScan at EBI, converted and merged to GOs. GO annotation can be modulated from Annex, GOSlim web services and manual editing. EC and KEGG annotations are generated from GO. Visual tools include sequence color code, KEGG pathways, and GO graphs with node highlighting and filtering options. Additional annotation data-mining tools include statistical charts and gene set enrichment analysis functions.

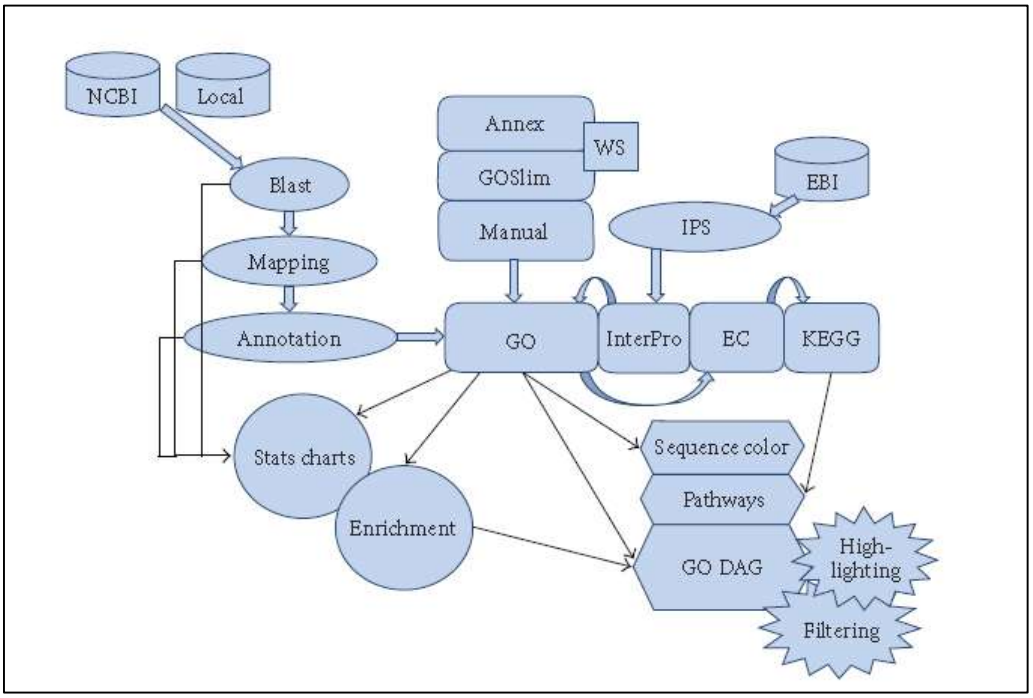


Figure 3: Schematic Representation of Blast2GO Application

The Blast2GO annotation procedure consists of three main steps: blast to find homologous sequences, mapping to collect GO terms associated to blast hits, and annotation to assign trustworthy information to query sequences.

Blast Step

The first step in B2G is to find sequences similar to a query set by blast. B2G accepts nucleotide and protein sequences in FASTA format and supports the four basic blast programs (blastx, blastp, blastn, and tblastx). Homology searches can be launched against public databases such as (the) NCBI nr using a query-friendly version of blast (QBlast). This is the default option and in this case, no additional installations are needed. Alternatively, blast can be run locally against a proprietary FASTA-formatted database, which requires a working www-blast installation. The Make Filtered Blast-GO-BD function in the Tools menu allows the creation of customized databases containing only GO annotated entries, which can be used in combination with the local blast option. Other configurable parameters at the blast step are the expectation value (e-value) threshold, the number of retrieved hits, and the minimal alignment length (hsp length) which permits the exclusion of hits with short, low e-value matches from the sources of functional terms. Annotation, however, will ultimately be based on sequence similarity levels as similarity percentages are independent of database size and more intuitive than e-values. Blast2GO parses blast results and presents the information for each sequence in table format. Query sequence descriptions are obtained by applying a language processing algorithm to hit descriptions, which extracts informative names and avoids low content terms such as “hypothetical protein” or “expressed protein”.

Mapping Step

Mapping is the process of retrieving GO terms associated to the hits obtained after a blast search. B2G performs three different mappings as follows.

a. Blast result accessions are used to retrieve gene names (symbols) making use of two mapping files provided by NCBI (geneinfo, gene2accession). Identified gene names are searched in the species-specific entries of the gene product table of the GO database.

b. Blast result GI identifiers are used to retrieve UniProt IDs making use of a mapping file from PIR (Non-redundant Reference Protein database) including PSD, UniProt, Swiss-Prot, TrEMBL, RefSeq, GenPept, and PDB.

c. Blast result accessions are searched directly in the DBXRef Table of the GO database.

Annotation Step

This is the process of assigning functional terms to query sequences from the pool of GO terms gathered in the mapping step. Function assignment is based on the gene ontology

vocabulary. Mapping from GO terms to enzyme codes permits the subsequent recovery of enzyme codes and KEGG pathway annotations. The B2G annotation algorithm takes into consideration the similarity between query and hit sequences, the quality of the source of GO assignments, and the structure of the GO DAG. For each query sequence and each candidate GO term, an annotation score (AS) is computed (see Figure 8). The AS is composed of two terms. The first, direct term (DT), represents the highest similarity value among the hit sequences bearing this GO term, weighted by a factor corresponding to its evidence code (EC). A GO term EC is present for every annotation in the GO database to indicate the procedure of functional assignment.

$$DT = \max(similarity \times EC_{weight})$$
$$AT = (\#GO - 1) \times GO_{weight}$$
$$AR : lowest.node(AS(DT + AT) \geq threshold)$$

Figure 4: Blast2GO Annotation Rule

ECs vary from experimental evidence, such as inferred by direct assay (IDA) to unsupervised assignments such as inferred by electronic annotation (IEA). The second term (AT) of the annotation rule introduces the possibility of abstraction into the annotation algorithm. Abstraction is defined as the annotation to a parent node when several child nodes are present in the GO candidate pool. This term multiplies the number of total GOs unified at the node by a user defined factor or GO weight (GOw) that controls the possibility and strength of abstraction. When all ECw's are set to 1 (no EC control) and the GOw is set to 0 (no abstraction is possible), the annotation score of a given GO term equals the highest similarity value among the blast hits annotated with that term. If the ECw is smaller than one, the DT decreases and higher query-hit similarities are required to surpass the annotation threshold. If the GOw is not equal to zero, the AT becomes contributing and the annotation of a parent node is possible if multiple child nodes coexist that do not reach the annotation cutoff. Default values of B2G annotation parameters were chosen to optimize the ratio between annotation coverage and annotation accuracy. Finally, the AR selects the lowest terms per branch that exceed a user-defined threshold.

Blast2GO includes different functionalities to complete and modify the annotations obtained through the above-defined procedure. Enzyme codes and KEGG pathway annotations are generated from the direct mapping of GO terms to their enzyme code equivalents. Additionally, Blast2GO offers InterPro searches directly from the B2G interface. B2G launches sequence queries in batch, and recovers, parses, and uploads InterPro results. Furthermore, InterPro IDs can be mapped to GO terms and merged with blast-derived GO annotations to provide one integrated annotation result. In this process, B2G ensures that only the lowest term per branch remains in the final annotation set, removing possible parent-child relationships originating from the merging action.

References

Altschul S. F., W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, “Basic local alignment search tool,” Journal of Molecular Biology, vol. 215, no. 3, pp. 403–410, 1990.

Ashburner M., C. A. Ball, J. A. Blake, et al., “Gene Ontology: tool for the unification of biology. The Gene Ontology Consortium,” Nature Genetics, vol. 25, no. 1, pp. 25–29, 2000.

Conesa and S. Gotz, "Blast2GO: A Comprehensive Suite for Functional Analysis in Plant Genomics," International Journal of Plant Genomics, vol. 2008, 2008.

Conesa, S. Gotz, J. M. Garcia-Gomez, J. Terol, M. Talon, and M. Robles, “Blast2GO: a universal tool for annotation, visualization and analysis in functional genomics research,” Bioinformatics, vol. 21, no. 18, pp. 3674–3676, 2005.

Myhre S., H. Tveit, T. Mollestad, and A. Lægreid, “Additional Gene Ontology structure for improved biological reasoning,” Bioinformatics, vol. 22, no. 16, pp. 2020–2027, 2006.

Ogata H., S. Goto, K. Sato, W. Fujibuchi, H. Bono, and M. Kanehisa, “KEGG: Kyoto Encyclopedia of Genes and Genomes,” Nucleic Acids Research, vol. 27, no. 1, pp. 29–34, 1999.

Ruepp, A. Zollner, D. Maier, et al., “The FunCat, a functional annotation scheme for systematic classification of proteins from whole genomes,” Nucleic Acids Research, vol. 32, no. 18, pp. 5539–5545, 2004.

Schomburg, A. Chang, C. Ebeling, et al., “BRENDA, the enzyme database: updates and major new developments,” Nucleic Acids Research, vol. 32, Database issue, pp. D431–D433, 2004.

Tatusov R. L., N. D. Fedorova, J. D. Jackson, et al., “The COG database: an updated version includes eukaryotes,” BMC Bioinformatics, vol. 4, p. 41, 2003.

Watson J.D., R.M. Myers, A.A. Caudy and J.A. Witkowski, "Recombinant DNA: Genes and Genomes - A Short Course," 3rd Ed., 2007.

Genome Assembly and Anotation

(Practical)

Bioinformatics analysis pipeline

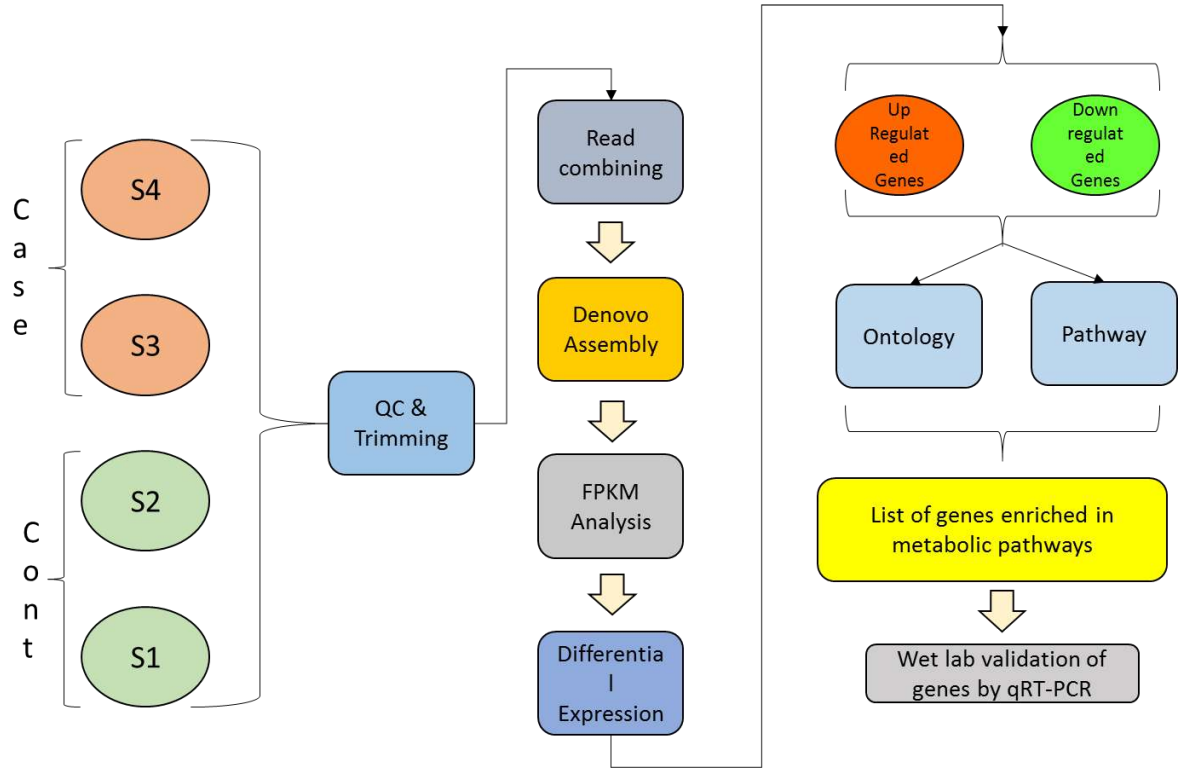


Figure 1: Schematic workflow of Transcriptome analysis

1.1 Read quality check - We check the following parameters from fastq file

- Base quality score distribution
- Sequence quality score distribution
- Average base content per read
- GC distribution in the reads
- PCR amplification issue
- Check for over-represented sequences

  ./Fastqc reads

Trimming & Contamination Removal - Based on quality of sequence reads, we trimmed sequence read where necessary, to retain only high quality sequence for further analysis. In addition, the low-quality sequence reads were excluded from the analysis from the Trimmed paired-end reads, we removed unwanted sequences. Adapter sequences and others. The trimming and contamination removal step had been performed using the help of Trimmimatic version Trimmomatic-0.35.

*java -jar trimmomatic-0.35.jar PE -phred33 input_forward.fq.gz input_reverse.fq.gz reads_1.fq.gz output_forward_unpaired.fq.gz reads_2.fq.gz output_reverse_unpaired.fq.gz ILLUMINACLIP:TruSeq3-PE.fa:2:30:10 LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15 MINLEN:3*6

Read assembly- The pre-processed high quality forward and reverse reads from each files were then assembled denovo using Trinity release v2.0.6 and output saved in file Trinity.fasta.

Trinity --seqType fq --max_memory 50G --left reads_1.fq.gz --right reads_2.fq.gz --CPU 6

Statistics of Assembled File

TRINITY_HOME/util/TrinityStats.pl trinity_out_dir/Trinity.fasta.

FPKM Analysis- Unigenes produced in Read assembly steps from Trinity.fasta were used as reference against reads from all four samples. RSEM tool was used to calculate the FPKM values for each unigene.

TRINITY_HOME/util/align_and_estimate_abundance.pl—seqType fq
--left left.fq --right right.fq—transcripts Trinity.fasta --output_prefix Sp_ds
--est_method RSEM --aln_method bowtie --trinity_mode --prep_reference
$TRINITY_HOME/util/abundance_estimates_to_matrix.pl --est_method RSEM
--out_prefix Trinity_trans Sp_ds.isoforms.results
Sp_hs.isoforms.results

DGE Analysis- Running Differential Expression Analysis

Edgar: http://bioconductor.org/packages/release/bioc/html/edgeR.html

DESeq2: http://bioconductor.org/packages/release/bioc/html/DESeq2.html

% R

> source("http://bioconductor.org/biocLite.R")

> biocLite('edgeR')

> biocLite('limma')

> biocLite('DESeq2')

> biocLite('ctc')

> biocLite('Biobase')

> install.*packages('gplots')*

> install.packages('ape')

$TRINITY_HOME/Analysis/DifferentialExpression/run_DE_analysis.pl --matrix counts.matrix --method edgeR --dispersion 0.1

FPKM values obtained from samples were combined in tabulated format and analyzed using TMeV4 tool to visualized differentially expressed genes, upregulated and down regulated genes are filtered on the bais of FC (fold change).

Annotation- blastx against Viridiplantae database (https://www.uniprot.org/uniprot/?query=taxonomy:33090) and futher uniprot id or ensemble ID of hit transcript can be used for online annotation server database.

Ontology Analysis- All the mapped ids were then submitted to *AgriGO/DAVID* tool for Ontology analysis to identify the Biological Process, Cellular Component, and Molecular Function in differentially expressed genes.

Biomolecular Sequence Encoding
(Practical)

The sequence data cannot be directly used as input in the Machine Learning Models for classification and prediction purposes. Hence, the encoding schemes are used to generate the numeric feature vector forms of sequences. Various encoding approaches used in this study are described below:

Amino acid Composition (AC)

The AC is the easiest and popularly used encoding method for representing the protein/peptide sequences. It is the fraction of each type of amino acid present in a peptide or protein sequence.

Di-peptide Composition (DC)

As there are 20 amino acids there can be 400 (20^2) possible di-peptide combinations. DC considers the amino acid ordering effect within a small range. Here, the fraction of each types of di-peptide to the total number of di-peptides in the sequence is calculated.

Tri-peptide Composition (TC)

The TC generates 8000 (20^3) descriptors and here the fraction of a tri-peptide in the sequence. Though this feature has been reported to give considerable accuracy but its computation and training is time consuming.

Amino acid anchoring Pair Composition (APC)

The APC features are generated by finding out the proportions of amino acid pairs separated by *g* residues where g= 0,1, 2 ...L_i-2 and L_i is the length of the sequence. It results a numeric feature vector of length *400*(g+1)*.

Composition-Transition-Distribution (CTD) features

CTD features describes amino acids generally based on properties like, hydrophobicity, polarity, polarizability and under each property there are 3 groups into which 20 amino acids are classified. The CTD features are:

i. Composition(C) of amino acids of a particular property (such as hydrophobicity) divided by the sequence length.
ii. Transition (T) that exemplify the percent frequency with which a class of amino acids with a specific property is followed by another class of amino acids having a different property.

iii. Distribution(D) determines the sequence length where the first, 25, 50, 75 and 100 percent of residues of certain characteristics are located.

Auto-Correlation Features (AF)

Auto-correlation considers the dependencies among the sequence features calculated based on the distribution of amino acid properties on a bio-molecular sequence. The properties of amino acids used to extract the AF are based on several types of amino acid indices available in AAindex Database (http:://www.genome.jp/dbget/aaindex.html). The following three types of autocorrelation descriptors will be used to encode the sequences.

i. Normalized Moreau-Broto Autocorrelation Descriptors (MB)
ii. Moran Autocorrelation Descriptors (MA)
iii. Geary Autocorrelation Descriptors (GA)

Conjoint triad descriptors (CT)

The CT descriptors (Shen *et al.*, 2007) consider the properties of an amino acid residue along with the residues preceding and succeeding it. As it considers three consecutive amino acids as a single unit, it is called as conjoint triad. The encoding scheme involves the clustering of 20 amino acid residues into seven classes based on their dipoles and side chain volumes. Hence, it generates a numeric feature vector of length 7^3 (343). It treats the triads equally those belong to the same class.

BLOSUM 62 descriptor (BL62)

The BL62 descriptors are derived from the BLOSUM substitution matrix for sequence similarity. These descriptors suggest the evolutionary significance of the epitope sequences.

R codes for encoding

```
setwd("—YOUR WORKING DIRECTORY--")
library(Biostrings)
library(protr)
library(BioSeqClass)
calculate<- function(kk, len, V1)
{
#Amino acid Composition
za <- matrix(nrow=len, ncol=20)
for(i in 1:len){
m <- as.character(kk[i])
km <- extractAAC(m)
za[i,]<- as.matrix(km, nrow=1, ncol=20)
```

```
}
za<-data.frame(cbind(V1,za))
#Dipeptide
zd <- matrix(nrow=len, ncol=400)
for(i in 1:len){
m <- as.character(kk[i])
km <- extractDC(m)
zd[i,]<- as.matrix(km, nrow=1, ncol=400)
}
zd<-data.frame(cbind(V1,zd))
#Tripeptide
ztc <- matrix(nrow=len, ncol=8000)
for(i in 1:len){
m <- as.character(kk[i])
km <- extractTC(m)
ztc[i,]<- as.matrix(km, nrow=1, ncol=8000)
}
ztc<-data.frame(cbind(V1,ztc))
##Amino acid anchoring Pair Composition (APC)
zgpc <- matrix(nrow=len, ncol=1200)
for(i in 1:len){
m <- as.character(kk[i])
km <- featureCKSAAP(m, 2)/rep(c(nchar(m), nchar(m)-1, nchar(m)-2), each=400)
zgpc[i,]<- as.matrix(km, nrow=1, ncol=1200)
}
zgpc<-data.frame(cbind(V1,zgpc))
#Autocorelation Moreaubroto
mb <- matrix(nrow=len, ncol=120)
for(i in 1:len){
m <- as.character(kk[i])
km <- extractMoreauBroto(m, nlag = 15L)
mb[i,]<- as.matrix(km, nrow=1, ncol=120)
}
mb<-data.frame(cbind(V1,mb))
#Autocorelation Moran
mr <- matrix(nrow=len, ncol=120)
for(i in 1:len){
m <- as.character(kk[i])
km <- extractMoran(m, nlag = 15L)
mr[i,]<- as.matrix(km, nrow=1, ncol=120)
```

```
}
mr<-data.frame(cbind(V1,mr))
#Autocorelation Geary
gr <- matrix(nrow=len, ncol=120)
for(i in 1:len){
m <- as.character(kk[i])
km <- extractGeary(m, nlag = 15L)
gr[i,]<- as.matrix(km, nrow=1,ncol=120)
}
gr<-data.frame(cbind(V1,gr))
#CTDC
ctdc <- matrix(nrow=len, ncol=21)
for(i in 1:len){
m <- as.character(kk[i])
km <- extractCTDC(m)
ctdc[i,]<- as.matrix(km, nrow=1, ncol=21)
}
ctdc<-data.frame(cbind(V1,ctdc))
#CTDT
ctdt <- matrix(nrow=len, ncol=21)
for(i in 1:len){
m <- as.character(kk[i])
km <- extractCTDT(m)
ctdt[i,]<- as.matrix(km, nrow=1, ncol=21)
}
ctdt<-data.frame(cbind(V1,ctdt))
#CTDD
ctdd <- matrix(nrow=len, ncol=105)
for(i in 1:len){
m <- as.character(kk[i])
km <- extractCTDD(m)
ctdd[i,]<- as.matrix(km, nrow=1, ncol=105)
}
ctdd<-data.frame(cbind(V1,ctdd))
#CTriad
ctr <- matrix(nrow=len, ncol=343)
for(i in 1:len){
m <- as.character(kk[i])
km <- extractCTriad(m)
ctr[i,]<- as.matrix(km, nrow=1, ncol=343)
```

```
}
ctr<-data.frame(cbind(V1,ctr))
#Blosum62
blosum62 <- matrix(nrow=len, ncol=175)
for(i in 1:len){
m <- as.character(kk[i])
km<- extractBLOSUM(m, submat = "AABLOSUM62", k = 5, lag = 7, scale = TRUE, silent = FALSE)
blosum62[i,]<- as.matrix(km, nrow=1, ncol=175)
}
blosum62<-data.frame(cbind(V1,blosum62))
write.table(za, "aminoacid_comp.txt", sep="\t", row.names=F, col.names=F, quote=F, append=T)
write.table(zd, "dipeptide_comp.txt", sep="\t", row.names=F, col.names=F, quote=F, append=T)
write.table(ztc, "trip_comp.txt", sep="\t", row.names=F, col.names=F, quote=F, append=T)
write.table(zgpc, "anchor_pair.txt", sep="\t", row.names=F, col.names=F, quote=F, append=T)
write.table(mb, "auto_corr_moreaubroto.txt", sep="\t", row.names=F, col.names=F, quote=F, append=T)
#write.table(mr, "auto_corr_moran.txt", sep="\t", row.names=F, col.names=F, quote=F, append=T)
write.table(ctdc, "ctdc.txt", sep="\t", row.names=F, col.names=F, quote=F, append=T)
write.table(ctdt, "ctdt.txt", sep="\t", row.names=F, col.names=F, quote=F, append=T)
write.table(ctdd, "ctdd.txt", sep="\t", row.names=F, col.names=F, quote=F, append=T)
write.table(ctr, "ctriad.txt", sep="\t", row.names=F, col.names=F, quote=F, append=T)
write.table(blosum62, "blosum62.txt", sep="\t", row.names=F, col.names=F, quote=F, append=T)
}
kkp <- readAAStringSet("__YOUR FILE NAME IN FASTA FORMAT___")
lenp <- length(kkp)
calculate(kkp, lenp, "Y")
```

Genome Editing to Epigenome Editing: A Newer Perspective in Crop Improvement

Introduction

The global human population is estimated to exceed 9 billion by 2050, which would require a predicted 70% increase in food production (Kumar, 2013). Moreover, one of the important challenges would be to produce the nutritive food from the continuously reducing per capita arable land and water. Another important challenge would be to produce this in a safe and sustainable manner (Kumar and Singh, 2014; Kumar, 2015a). The conventional approaches might not be adequate to meet the projected food requirements, both in terms of quantity and quality. Since most of the cultivated varieties have reached their yielding plateau, the need of the day is to deploy modern tools and techniques to further enhance the productivity of crop plants with the decreasing availability of natural resources. Importantly, the biosafety issues of genetically modified organisms (GMOs), particularly those associated with the genetic manipulation technology being used (Kumar et al., 20006; Kumar, 2014), have become a serious concern. Therefore, appropriate safety guidelines framed/being framed by the regulatory agencies of the country must be followed for environmental safety (Kumar, 2012; Kumar, 2015b). Fortunately, epigenome editing promises to provide unprecedented opportunities not only for the manipulation of biological systems for a better understanding of the regulatory mechanisms but also for efficient manipulation of the genome to improve stress tolerance against the climatic changes. This would enable functional integration of epigenetic marks and their usage towards improving the agriculturally important traits in the crop plants (Springer and Schmitz, 2017).

Plant scientists aim at developing newer crop varieties with wider adaptation to the changing climatic conditions. Researchers have been interested in deciphering the underlying mechanisms that enable plants to better adapt to diverse environmental conditions. Increasing knowledge of the plant's genome structure and functional characterizations have paved the way not only for the selection of parental/recombinant lines for crossing but also for targeted genome and epigenome editing leading towards the so-called next-generation biotechnology for their application in breeding for crop improvement. Technological advancements in the manipulation of DNA/gene have enabled us to better understand and utilize the genome. However, site-specific manipulation in the genome of an organism has been elusive for quite a long time. Identification, isolation and stable integration of a gene of interest in the plant of our choice has been successful to a great extent, but consistency in the performance of the transgenic plants and the related biosafety issues have been significant points of concerns. Moreover, there have been several technical issues associated with the tools and techniques

used for the development of GMOs, such as the copy number of the transgene, site of integration of transgene, presence of selectable marker gene(s) in the transgenic plant, expression level/stability of the transgene, etc. To overcome many of these challenges, scientists have been working day and night to bring innovations in genetic manipulation technologies.

Genome editing: a continuously improving technology

Targeted gene editing has emerged as an alternative to the standard genetic manipulation methods for crop improvement. This has been possible due to the advances in engineering the nucleases with programmable, site-specific DNA-binding domains like zinc finger nucleases (ZFN), transcription activator-like effector nucleases (TALENs) and Clustered regularly interspaced short palindromic repeats/CRISPR-associated-9 nuclease (CRISPR/Cas9). Each of these gene-editing tools has its own advantages and limitations. However, CRISPR/Cas9, a newer method based on the bacterial CRISPR and Cas9 type II prokaryotic adaptive immune system, has emerged as a simpler, easier, and more precise/successful tool for genome editing. Originally identified in *Streptococcus pyogenes*, the CRISPR/Cas9-mediated double-strand breakage relies upon two interacting RNA moieties: (i) CRISPR RNAs (crRNA), and (ii) trans-activating RNAs (tracrRNA) for sequence specificity. When it was demonstrated that a single chimeric RNA molecule comprising of these two RNAs can serve the function of recruiting Cas9 nuclease, its usage in genome editing in a site-specific manner became easier. Any sequence (~20 nucleotides long) in the genome can be a target, provided it meets the two basic requirements: (i) the sequence is unique within the genome, and (ii) the target sequence is located immediately upstream of a Proto-spacer Adjacent Motif (PAM). The PAM sequence is essential for target sequence identification/binding. The Cas9 protein and the guide-RNA (gRNA) form a riboprotein complex. Once the gRNA-Cas9 complex is formed, Cas9 undergoes a conformational change from an inactive (non-DNA binding confirmation) to an active (DNA-binding) form. Importantly, the 'spacer' sequence of gRNA remains free to interact with the target DNA. The Cas9-gRNA complex can bind at any genomic sequence having the PAM, but the extent to which the gRNA spacer complements with the target DNA determines whether Cas9 will make cut or not. The Cas9 nuclease, having two functional endonuclease domains (i) RuvC and (ii) HNH, undergoes another conformational change upon binding at the target DNA, which positions the nuclease domain to cleave opposite strands of the target DNA causing double-strand break (DSB) at the target (~ 3–4 nucleotides upstream of the PAM sequence. Subsequently, the resulting DSB is repaired by one of the two repair mechanisms: (i) an efficient but error-prone Non-Homologous End Joining (NHEJ) pathway, and (ii) a less efficient but high-fidelity Homology Directed Repair (HDR) pathway (Fig. 1).

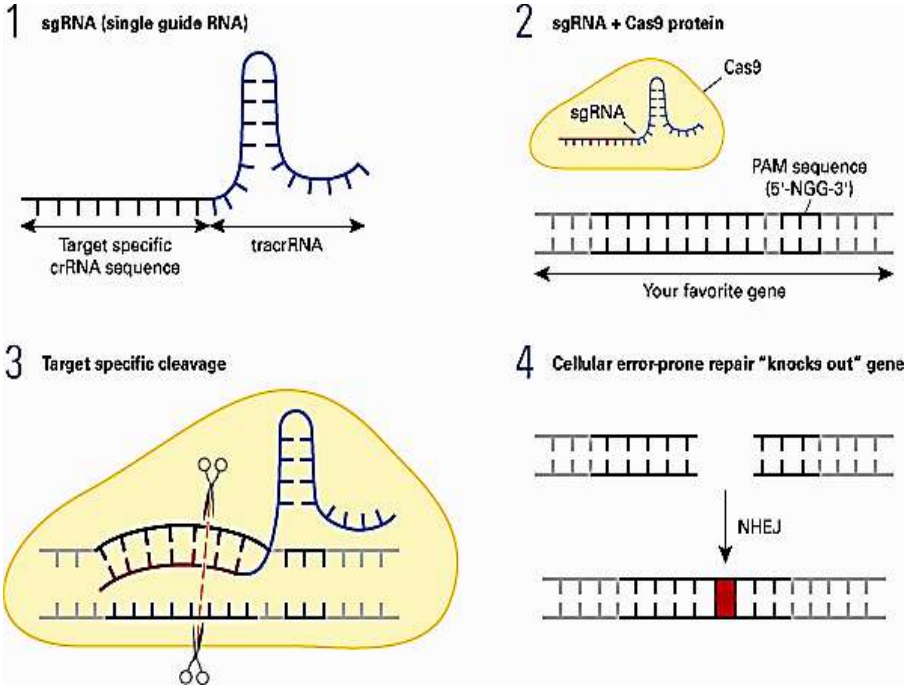


Figure 1. Steps in CRISPR/Cas9-mediated gene editing. 1. The two components (i) CRISPR RNAs (crRNA), and (ii) trans-activating RNAs (tracrRNA) of a single-guide RNA (sgRNA). 2. The sgRNA-Cas9 complex, and the target DNA (gene) containing the PAM sequence. 3. Target-specific cleavage of the DNA through binding/recognition of the PAM sequence, and double-strand break (DSB) at the target DNA. 4. Repair of the break using an efficient but error-prone non-homologous end joining (NHEJ) mechanism.

Recent advances in genome editing technology

The discovery of a catalytically-dead mutant of *Streptococcus pyogenes* SpCas9 (dCas9) has provided a worthy tool for regulating gene expression (Qi et al., 2013). Moreover, novel tagging approaches have enabled more efficient recruitment of multiple effectors through a single-dCas9 to a specific genomic locus. The recruitment strategy has also been combined with a chemically inducible approach to achieve temporal control of transcriptional regulation. These advances focus on the recruitment of synthetic modulators and the reversibility required for mechanistic studies. Braun et al. (2017) used FIRE–Cas9 for rapid and reversible recruitment of endogenous chromatin complex to a genomic locus in a cell. While the previously known strategies recruit exogenous activators/repressors to turn on and off the gene expression in the cells cultured for several days, use of induced proximity (synthetic ligands) enables to determine the link between epigenetic regulators and histone modifications within minutes of recruitment.

Alternatives to Cas9 nuclease are being searched by the scientist for a more effective and precise nuclease targeting different nucleic acids. Cpf1 (a CRISPR endonuclease discovered in Prevotella and Francisella 1 bacteria), is such an alternative platform for CRISPR-based

genome editing. CRISPR–Cpf1 system enhances genome-editing efficiency and speed. Cpf1 is also known as Cas12a it is more effective and precise compared to the Cas9. A newly discovered nuclease namely Cas14 possesses a single-stranded DNA targeting activity having two times smaller size than Cas9. Hence, it can be potentially utilized for detecting ssDNA viruses of clinical, ecological, and economic importance. Its non-specific ssDNase cleavage activity can also be combined with isothermal amplification method for its use in high-fidelity DNA single-nucleotide polymorphism genotyping. Moreover, Cox et al. (2017) reported the possibility of editing RNA transcripts to alter their coding potential in a programmable manner. The RNA Editing for Programmable A to I Replacement (REPAIR), a transcriptome-editing technology (targeting and altering RNA bases) offers an opportunity to even edit the mRNA. Cox et al. (2017) used PspCas13b enzyme in their REPAIR technology for both RNA knockdown and RNA editing having broad applicability for biotechnology research and therapeutics. RNA editing would allow answering some of the questions about alternative splicing, and translation. RNA editing would confer temporary, reversible genetic edits, rather than the permanent genome edits in case of DNA editing. This might allow avoiding the ethical issues that might arise around the genome editing. However, the RNA base editors would have to be administered repeatedly to be a functional therapeutic approach. Thus, gene/genome editing technology has a very promising future in the areas of research and therapeutics.

Epigenomics: an immerging area of functional genomics

Information needed for proper assembly of RNAs and proteins in any living organism is encoded in the cellular genome. But, the instructions regarding access to this information in a temporal and spatial manner is encrypted in the epigenome, which ultimately grants selective access to the information contained in the DNA/gene. Plants are sessile in nature and face multiple environmental stresses throughout their life. Although pants possess innate capability to tolerate such adverse climatic conditions, yet they require improvement in their efficiency to produce more under unfavourable climatic conditions. Until recently it has been thought that isolation of a gene associated with a trait would be sufficient enough to transfer the trait to a crop plant and to create the expected phenotype. However, evidence suggests that nucleotide sequence of the gene provides only part of the genetic information, the surrounding environment like chromatin confirmation also contributes to the expression of the trait. Since the epigenetic states of chromatin are variable, transfer of a trait from one species/plant to another would not only require the transfer of the gene(s) associated with the trait but also the appropriate chromatin/epigenetic states so that the trait can express under suitable epigenetic environment. It is, therefore, essential to study the epigenetic states of the donor plant/species and to ensure that proper re-establishment of the epigenetic state of the genes takes place in the recipient plant/species for appropriate expression of the trait (Kumar, 2019b).

While a sum total of all the genes in an organism is known as genome, epigenome refers to the sum total of all the epigenetic changes in DNA (without any alteration in the underlying

nucleotide sequence) and/or in the structural components of the genetic material that affect expression/activity of the gene/genome. Epigenetics is the study of such variations affecting gene expression in the cell/organism (Kumar 2018b). Epigenetic changes include methylation of cytosine resulting in the formation of 5-methylcytosine (5-mC) (Kumar et al. 2018), histone protein modifications, variation in the biogenesis of small-RNA (sRNA) (Wang et al. 2016). Growing evidence indicates the involvement of epigenetic regulation during the developmental processes as well as during biotic and abiotic stresses in plants and animals. Epigenetic changes may revert back to the original state soon after normalization of the conditions. Interestingly, some (~30%) of the epigenetic changes may be carried over the next generation that often results in phenotypic variations (Kumar, 2018b). Thus, it is has become evident that epigenetic changes play important roles in acclimatization, stress tolerance, adaptation, and evolutionary processes in living organisms (Kumar, 2019a). Therefore, it is important to discover the epigenetic machinery of gene regulation for crop improvement towards the development of climate-smart crop plants to meet the challenges of food and nutritional security for the global population. Since the rates of genetic mutations and phenotypic variations are considerably different, they cannot be explained merely based on genetics. Additional machinery such as epigenetics can help explaining this enigma (Kumar, 2017). If epigenetics is considered a complementary mechanism, many of the phenotypic variations (e.g. dissimilarity between the clones) can be easily explained.

It has been reported that the rate of spontaneous epimutations is higher in the CG context because these sites are not retargeted by RdDM. DNA methylation generally refers to the addition of a methyl group at the 5^{th} carbon of cytosine as a post-replicative event (Figure 2). While CHH methylation is maintained by Domains Rearranged Methyltransferase 2 (DRM2), it is responsible for *de novo* methylation in all the contexts of cytosine at least in Arabidopsis. DRM2 is recruited to the target loci by a specialized 24 nucleotide small interfering RNA (RNA-directed DNA methylation pathway). Cytosine methylation homeostasis is determined by the DNA methylation and demethylation processes. Demethylation of the promoter and/or coding region may also be required to activate the expression of specific genes under the changing environmental conditions or during the developmental stages of a plant (Li et al., 2018). A variety of histone modifications and their possible combinations (like H3K4me3 & H3K27Ac: activation marks, and H3K9me3 & H3K27me3: repressive marks) affect the transcriptional potential of the gene. Histone methylation can also be reversed by the action of different types of histone demethylases. Studies also indicate that the genome-wide hypo/hyper-methylation induces biogenesis of 24-nt siRNAs, and activates *de novo* (de)methylation pathways. Recent studies reveal a highly cell type-specific nature of epigenetic regulation of genes which indicates the need for new technologies to study the functions of chromatin regulators in a cell-specific manner, at the specific developmental stage, and in proper genomic context. Hence, in-depth studies would be necessary to understand the role of the RdDM pathway and the chromatin regulators in epigenetic regulation of gene expression

and its deployment in epigenetic engineering of crop plants. However, epigenetic mechanisms of gene regulation are yet to be fully understood and utilized as epialleles (the alleles that are genetically identical but epigenetically different due to the epigenetic modifications, showing variable expression) in crop improvement programs.

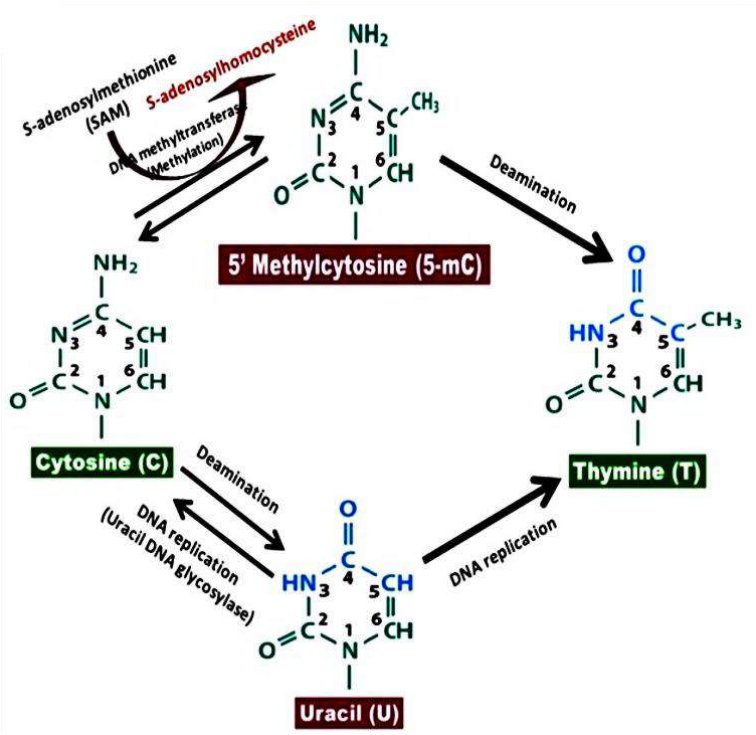


Figure 2. Epigenetic modifications of the genetic material. (**A**) Conversion of cytosine (C) into 5-methylcytosine (5-mC) and other bases, and its reversal/repair.

The conventional approaches may not be adequate to meet the projected food requirements, both in terms of quantity and quality. Moreover, most of the cultivated crops/varieties have reached the yield plateau. Therefore, the need of the day is to deploy modern tools and techniques to further enhance the productivity of crop plants, the nutritional quality of the product, and to explore the possibility of producing novel molecules (molecular farming) with the decreasing availability of natural resources. In view of the biosafety concerns of genetically modified organisms currently being associated with the genetic manipulation of crop plants (Kumar et al., 20006; Kumar, 2014), epigenetic engineering (supposed to have limited biosafety issues) would be a better approach (Kumar, 2018). However, appropriate safety guidelines framed by the regulatory agencies of the country must be followed for personnel, laboratory and environmental safety (Kumar, 2012; Kumar, 2015b). Thus, epigenome editing may provide unprecedented opportunities for the manipulation of biological systems in an efficient/effective manner to improve stress tolerance against climatic conditions.

Epigenome editing: possibilities and future perspectives

Epigenome editing is a very promising approach that can usher a new era for novel applications of basic research and molecular medicine. Epigenetic editing is based on fusion proteins comprising a designed DNA recognition domain that targets an attached enzymatic domain to defined genomic target sites. Because the target recognition of CRISPR/Cas9 complex is based on Watson/Crick base-pairing between a guide RNA and one DNA strand of the target site, re-targeting of CRISPR/Cas9 only requires the introduction of a new guide RNA sequence. Histone acetylation and deacetylation play an important role in regulation of gene expression. dCas9 fused with HDAC3 protein can function as a synthetic histone-deacetylase to modulate gene expression. Different groups of scientists are currently working world over to identify the gene(s) involved in epigenetic changes to establish proof of the concept of epigenetic manipulation in plant. However, many areas of epigenetics remain to be explored. We still know only a little about the factors that regulate targeting of active DNA demethylation during developmental stages. Does DNA (de)methylation interplay with other epigenetic features or chromatin features? Future research should aim at identifying more developmental processes in different species that involve epigenetic regulation. Assessing the contribution of transgenerational epimarks to heritable phenotypic variation has been a major challenge as many of the chromatin (DNA methylation and histone modification) changes and gene expression variants co-segregate with DNA sequence polymorphisms. Nonetheless, there is evidence that plants possess heritable epiallelic variations that can be associated with the trait of interest and utilized for crop improvement. We are still at the beginning of understanding the transgenerational stability of epigenetic variations. Only a little is known to us about the role of the environment in the creation of induced epialleles. We can anticipate that soon epigenome editing will provide a means to assess the role of a QTL in epiallelic variations which may provide an interesting new route for the improvement of crop plants. The proteins involved in DNA (de)methylation, histone modification and the mechanisms of ncRNA mediated regulation of developmental processes in plants are becoming clear day by day.

The discovery of dCas9 has provided a valuable tool for epigenome editing (Thakore et al., 2016). The recent studies have been focused on regulatory DNA sequences through the recruitment of dCas9 fused to the histone acetyltransferase, and Tet1 DNA demethylase to activate enhancers (Liu et al., 2016). Braun et al. (2017) used FIRE–Cas9 for reversible recruitment of endogenous chromatin complexes to any genomic locus in almost any cell type of mammalian system. The enzymes responsible for writing, erasing, and reading epigenetic marks are multi-protein complexes and becoming known day by day. By fusing a single subunit of a chromatin complex with a chemical-induced proximity tag, Frb (FKBP-rapamycin-binding domain of mTOR), Braun et al. (2017) could rapidly recruit intact multi-subunit complexes to a specific genomic sequence upon rapamycin (RAP) treatment. Locus specificity was obtained through the expression of a complementary dimerizer Fkbp (FK506-binding-protein) fused

with a dCas9–MS2 anchor. Focusing on the recruitment of Hp1/Suv39h1 heterochromatin complex and the BAF chromatin-remodeling complex, they could demonstrate possibilities of both gene repression and activation through epigenome editing. This provides new insight into the fine-tuning of epigenetic mechanisms. Recently, Fukushima et al. (2019) demonstrated in vivo epigenome editing using a new construct, dCas9-olEzh2 (Ezh2 from *Oryzias latipes* fused to dCas9) to manipulate H3K27me3. They showed that dCas9-olEzh2 accumulates H3K27me3 at the targeted loci which induced gene repression in Japanese Killifish (*Oryzias latipes*) embryos. These in vivo epigenome editing will be very useful for epigenetic regulation of gene expression and heritability of epigenetic modification at targeted genomic loci.

The views expressed here are those of the author only. These may not necessarily be the views of the Institution/Organization the author is associated with.

References

Braun SMG, Kirkland JG, Chory EJ, Husmann D, Calarco JP, Crabtree GR (2016) Rapid and reversible epigenome editing by endogenous chromatin regulators. Nature Communication 8: 560. doi: 10.1038/s41467-017-00644-y.

Cox DBT, Gootenberg JS, Abudayyeh OO, Franklin B, Kellner MJ, Joung J, et al. (2017). Science 358, 1019–1027. doi: 10.1126/science.aaq0180.

Fukushima HS, Takeda H, Nakamura R (2019) Targeted in vivo epigenome editing of H3K27me3. Epigenetics Chromatin 12:17. https://doi.org/10.1186/s13072-019-0263-z.

Kumar S (2012) Biosafety issues in laboratory research. Biosafety 1: e116. doi:10.4172/2167-0331.1000e116.

Kumar S (2013) The role of biopesticides in sustainably feeding the nine billion global populations. J Biofertil Biopestici 4: e114.

Kumar S (2014) Biosafety issues of genetically modified organisms. Biosafety 3: e150. doi:10.4172/2167-0331.1000e150.

Kumar S (2015a) Biopesticide: An environment friendly pest management strategy. J Biofertilizer Biopesticide 6: doi:10.4172/2155-6202.1000e127.

Kumar S (2015b) Biosafety and biosecurity issues in biotechnology research. Biosafety 4: e153. doi:10.4172/2167-0331.1000e153.

Kumar S (2017) Epigenetic control of apomixis: a new perspective of an old enigma. Advances Plants Agriculture Research 7: 00243.

Kumar S (2018a) Epigenomics of plant responses to environmental stress. Epigenomes 2: 6. doi: 10.3390/epigenomes2010006.

Kumar S (2018b) Environmental stress, food safety, and global health: biochemical, genetic and epigenetic perspectives. Medical Safety Global Health 7: e145

Kumar S (2018c) Epigenetic memory of stress responses in plants. J Phytochemistry Biochemistry 2: e102.

Kumar S (2019a). Epigenetics and epigenomics for crop improvement: Current opinion. Advances Biotechnology Microbiology 14: 555879. doi: 10.19080/AIBM.2019.14.555879

Kumar S (2019b) Epigenomics for crop improvement: Current status and future perspectives. J Genetics Cell Biology 2: 1–6.

Kumar S, Arul L, Talwar D, Raina SK (2006) PCR amplification of minimal gene expression cassette: an alternative, low cost and easy approach to 'clean DNA' transformation. Current Science 91: 930–934.

Kumar S, Beena AS, Awana M, Singh A (2017b) Salt-induced tissue-specific cytosine methylation downregulates expression of *HKT* genes in contrasting wheat (*Triticum aestivum* L.) genotypes. DNA Cell Biology 36: 283–394. doi:10.1089/dna.2016.3505.

Kumar S, Beena AS, Awana M, Singh A (2017c) Physiological, biochemical, epigenetic and molecular analyses of wheat (*Triticum aestivum*) genotypes with contrasting salt tolerance. Frontiers Plant Science 8: 1–20. doi:10.3389/fpls.2017.01151.

Kumar S, Chinnusamy V, Mohapatra T (2018) Epigenetics of modified DNA bases: 5-methylcytosine and beyond. Frontiers Genetics 9: 1–14.

Kumar S, Krishnan V (2017) Phytochemistry and functional food: The needs of healthy life. J Phytochemistry Biochemistry 1: 103.

Kumar S, Sahu N, Singh A (2015) High-frequency in vitro plant regeneration via callus induction in a rare sexual plant of *Cenchrus ciliaris* L. In Vitro Cellular Developmental Biology–Plant 51: 28–34.

Kumar S, Singh A (2014) Biopesticides for integrated crop management: Environmental and regulatory aspects. J Biofertilizers Biopesticides 5, e121.

Kumar S, Singh A (2016) Epigenetic regulation of abiotic stress tolerance in plants. Advances Plants Agriculture Research 5: e00179.

Kumar S, Singh A 2016. Epigenetic regulation of abiotic stress tolerance in plants. Advances Plants Agriculture Research 5: 00179. doi: 10.15406/apar.2016.05.00179.

Kumar S, Singh AK, Mohapatra T (2017a) Epigenetics: history, present status and future perspective. Indian J Genetics Plant Breeding 77: 445–463.

Li Y, Kumar S, Qian W (2018) Active DNA demethylation: mechanism and role in plant development. Plant Cell Reports 37: 77–85. doi: 10.1007/s00299-017-2215-z.

Liu XS et al. (2016) Editing DNA methylation in the mammalian genome. Cell 167: 233–247.

Qi LS et al. (2013) Repurposing CRISPR as an RNA-guided platform for sequencespecific control of gene expression. Cell 152: 1173–1183.

Springer NM, Schmitz RJ (2017) Exploiting induced and natural epigenetic variation for crop improvement. Nature Review Genetics doi:10.1038/nrg.2017.45

Thakore PI, Black JB, Hilton IB, Gersbach CA (2016) Editing the epigenome: technologies for programmable transcription and epigenetic modulation. Nature Methods 13, 127–137.

Wang X, Li Q, Yuan Q, Kumar S, Li Y, Qian W (2016) The cytosolic Fe-S cluster assembly component MET18 is required for the full enzymatic activity of ROS1 in active DNA demethylation. Scientific Reports 6: 26443. doi: 10.1038/srep26443.

Machine Learning Techniques

Machine Learning

Learning denotes changes in a system that enable a system to do the same task more efficiently the next time or Learning is constructing or modifying representations of what is being experienced. Machine learning is a scientific discipline concerned with the design and development of algorithms that allow computers to evolve behaviours based on empirical data, such as from sensor data or databases. A learner can take advantage of examples (data) to capture characteristics of interest of their unknown underlying probability distribution. Data can be seen as examples that illustrate relations between observed variables. Discover new things or structure that is unknown to humans eg. data mining. A major focus of machine learning research is to automatically learn to recognize complex patterns and make intelligent decisions based on data.

Types of Machine Learning

Broadly, machine learning is classified into two categories i.e. supervised and unsupervised learning. Supervised learning generates a function that maps inputs to desired outputs based on labelled training data, where the desired output for each object is known. Approaches of supervised learning are classification and prediction. The prevalent techniques of supervised learning are Naïve Bayes classifier, Logistic Regression, Linear Discriminant Analysis, K-Nearest-Neighbour classifiers, Artificial Neural Networks, Support vector machine etc.

Unsupervised learning discovers underlying patterns in the data based on unlabelled training data. In other words if data has to be processed by machine learning methods, where the desired output is not known, then the learning task is called unsupervised. Approaches to unsupervised learning include clustering (e.g., k-means, hierarchical clustering)

1. Supervised Machine Learning Techniques

Supervised Classification technique is based on the principles of machine learning techniques in which of parameters of inferring a function is estimated based on training data such that a set of input vector, which consists of realized values of explanatory factors is being used to get desired output values of dependant factors with desired accuracy. This function is also called as classifier. This inferred function is expected to predict correct output value for any valid input vector. This means, it requires the learning algorithm to generalize from the training data to unseen situations with desired accuracy. In order to develop reliable inferred function following steps needs to be followed:

- Selection of appropriate training data set which is to be representative of real world of the problem under consideration along with representative sets of output values.
- Selection of input features (factors) which can able to predict the output with desired accuracy but should not be too large in numbers.
- Determination of structure of the function and corresponding learning algorithms based on optimized performance through cross validation techniques on s sub-set of training data set which is also known as validation set. Certain control parameters are used for this purpose.
- Evaluation of the accuracy of the learned function after parameter adjustments on a test data set which is different from the training set.

Large numbers of supervised learning algorithms are available in literature with their advantages and disadvantages but there is no single algorithms which can be used on all types of data sets. There are four major issues which needs special consideration in supervised learning:

Tradeoff between bias and variance: The prediction error is sum of bias and variance of the learning algorithms. Generally it is desirable that a learning algorithm with low bias should be flexible such that it can fit the data set but it should not be that flexible that it fit differently to each training data set due to its high variance. Therefore, it is necessary to adjust this tradeoff between bias and variance.

Availability of dataset and complexity of function: In case, simple true function, learning algorithm with high bias and low variance will results reliable inferred function with the help of small amount of dataset. But in case of highly complex true function resulting from interactions within different components needs large amount of training dataset to build learning algorithm with low bias and high variance. Therefore, it is desirable for good learning algorithms to automatically adjust the bias/variance tradeoff based on the amount of data available and the apparent complexity of the function to be learned.

Dimensions of input dataset: Large dimension of the dataset may create confusion and it may become difficult learning problem even if the true function depends on only small number of features. This will results in large variance. Hence, high input dimensionality typically requires tuning the classifier to have low variance and high bias. It is always desirable to apply feature selection procedures or dimensionality reduction techniques to get desirable output.

Noisy output values: In case output values are incorrect beyond a limit due to response errors then the learning algorithm is expected to lead to undesirable inferred function .This is case where it is usually best to employ a high bias, low variance classifier.

The selection of learning algorithms depends on number of other factors such as (i) heterogeneity of the data, (ii) redundancy of data and (iii) linear and non-linear relationships among the factors etc. In case the input feature dataset is heterogeneous such as discrete, discrete ordered, counts, continuous values then decision tree based methods are easy to apply whereas learning algorithms, including support vector machines, linear regression, logistic regression, neural networks and nearest neighbor methods can be applied on numerical feature which are scaled to similar ranges (e.g., to the [-1,1] interval). If the input features contain redundant information in terms of multi-collinearity number of learning algorithms such as linear regression, logistic regression and distance based methods performs poorly because of numerical instabilities. The best way to solve this problem is through imposing regularization conditions. Again, if there are complex interactions among features then algorithms based on decision trees and neural networks work better due to their inherent capabilities to deal with this situation. The selection of the best algorithm can be done using cross validation techniques to the given dataset and problem at hand. Some of the most widely used learning algorithms are support vector machines, linear regression, logistic regression, naive Bayes, linear discriminant analysis, decision trees, k-nearest neighbor algorithm and Neural Networks (multilayer perception).

Classification and Prediction Techniques

Let the total sample size of the biological data set is "n" and there are "m" features on which data is available. Let, the data matrix of size m x n is represented by $\mathbf{X}=(x_{ij})$ where, x_{ij} represent data on *j*-th feature of i-th observation. Further, let $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ is a matrix of response variable which may be categorical in nature depending on number of classes for classification problem.

K-Nearest-Neighbor classifiers (K-NN): This classifier is based on the distances among closest K neighbors to a particular unit. If, sample i and j are represented by $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots\ldots\ldots, x_{im})'$ and $\mathbf{x}_j = (x_{j1}, x_{j2}, \ldots\ldots\ldots, x_{jm})'$ then distances can be calculated based on Euclidean distance technique on standardized data set. In order to classify new observation, majority vote among K- neighbors is being considered for its classification in a particular class. It has been observed that as K increases, the variance of the classifier decreases but its bias increases. It has been shown by Friedman 1997; Dudoit et al. (2002), that this classifier is highly consistent. Best results can be obtained when feature selection step is employed before application of the classifier.

Artificial Neural Networks: This technique was proposal by Barnard Widrow (1950). This is a data driven and non- parametric model based approach. In this, a network of nodes (neurons) is being generated through assigning different weights. This includes may be obtained through both supervised and unsupervised learning. The basic principle of learning in this case is modifications in synaptic weights which are determinant through learning algorithm. For

example in correlation learning rule; weights are adjusted according to Hebb's rule. $\Delta(W_{ij})= O_iO_j$, where ΔW_{ij} is change in weights of *i*-th node, which is connected to *j*-th node and O_i is output of *i*-th unit. In case of learning rule via error correction, weights are adjusted by minimizing output errors with respect to weights i. e. $\Delta(W_{ij})$=E(O_iO_j). This algorithm is being used in Perceptron, MADALAINE and back propagation models. This technique is capable of solving number of complex problems but it is complex and computationally extensive.

Classification Tree: The classification tree technique can be broadly classified in two categories i.e. (i) binary tree such as CART, QUEST etc. and (ii) multi-way split tree such as CRUISE. In order to gain classification accuracy, these techniques use discriminant based procedure for splitting. In a CART, node of the tree splitting recursively to make data homogeneous till the tree is fully grown. The impurity at node t can be measured by Gini diversity index of Breiman et al. 1984 as i(t) = $\sum_k p^2(k/t)$, where, p(k/t) is the probability that a sample is in class k given that it falls into node t. Two approaches are normally applied to avoid over fitting of data. First, is cross validation approach, with minimal cost-complexity pruning method of CART. Second, cross validation multistep look-ahead stopping rule to determine the proper depth of the tree.

Random Forest: This method has been proposed by Breiman (2001) and it is available in package form (Random Forest) in R software. It is based on bagging technique which is different from boosting. In this case, same training set is being used by each classifier and hard to classify sample observation get more weight, while in bagging which is based on bootstrapping technique, yields incomplete overlap training samples among classifiers with some sample get more weight randomly. This technique generates diversity by taking bootstrap samples for each tree generated (bagging). Also, it selects random sub-space of the predictor at each node. The number of feature selected at each node may also vary. However, default value of number of predictors which is taken is $\sqrt{m}$.

Support Vector Machine (SVM): This technique was introduced by Vapnik (1995). This technique is based on finding linear hyperplanes in input space and kernel space for avoiding over fitting. Let training sample data consist of n pairs ($\mathbf{x}_1, y_1$) , ($\mathbf{x}_2, y_2$) , ……. , ($\mathbf{x}_n, y_n$) with $\mathbf{x}_i \epsilon R^p$ and $y_i \epsilon$ {-1, 1} then SVM classifier finds hyperplane (P_0) bisecting closest points of the data which is linearly seperable. The P_0 is defined as

$$\{ \ : f(\mathbf{x}) = \mathbf{x}'\boldsymbol{\beta} + \beta_0 = 0\} \text{ and } \|\boldsymbol{\beta}\|=1$$

Classifier creates a parallel hyperplane P_1 such defined as

$$(P_1)\ \{\mathbf{x}' : f(\mathbf{x}') = \mathbf{x}'\boldsymbol{\beta} + \beta_0 = -1\}$$

On a point in class -1 closet to P_0 and second hyperplane P_2 as

$(P_2) = \{\mathbf{x} : f(\mathbf{x}) = \mathbf{x}\boldsymbol{\beta} + \beta_0 = 1\}$ on a point in class closet toP_0.

The optimum hyperplane for separating the data can be formed by maximizing the perpendicular distance between two parallel supporting planes P_1 and P_2 i.e. M. The resulting classifier can be given by

$$\hat{y} = \text{sign}\,(\mathbf{x}\boldsymbol{\beta} + \beta_0)$$

As we know that classes are separable. So, m = $2/\|\boldsymbol{\beta}\|$ there maximization of M leads to minimization of $\|\boldsymbol{\beta}\|/2$

Therefore, this problem can be reduced to minimization of

$$\emptyset(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\| / 2$$

Subject to y_i ($\mathbf{x}_i'\,\boldsymbol{\beta}$ + β_0) ≥ 1 for all { ($\mathbf{x}_i$, y_i) , i= 1,2,……,n }

In case data set is not separable then this technique maps the data into higher dimensional space where training set is separable via some transformation.

K : $\mathbf{x} \rightarrow \emptyset(\mathbf{x})$.

A kernel function K ($\mathbf{x}_i$, $\mathbf{x}_j$) = < $\emptyset$ ($\mathbf{x}_i$) , $\emptyset(\mathbf{x}_j)$ > computes inner product in some expanded feature space. Linear or Gaussian kernels are widely used.

Boosting Method: This method combines weak classifiers and takes weighted majority vote of their predictions. It has been developed for improving the performance of any weak learning algorithm. It changes adoptively the distribution of training data set based on performance of previous classifiers. Therefore, for combining weak classifiers, it takes weighted majority vote of the prediction. In this, weighted version of same data set has been used, where weight for difficult to classify observation is increased in subsequent classifier. Adaboost (Freund and Schapire 1997) is an algorithm which creates number of classifiers from same dataset and combines the tree using majority vote.

Naive Bayes Classifier: It is a simple probabilistic classifier based on Bayes theorem. It is also known as independent feature model due to its strong assumptions of independence. It based on the assumption that inclusion or exclusion of a particular feature in the model is independent of inclusion or exclusion of any other feature and their contributions towards probability is independent of each other. In many practical applications maximum likelihood method can be used in estimation of parameters of this model instead of Bayesian probability. Depending on the probability model it can be trained efficiently using supervised learning techniques. It has been found in literature that, in spite of its oversimplified assumptions it out performed number of current approaches such as boosted trees or random forest under specific situations. One of the major advantages of this classifier is that it requires limited amount of training data set for

estimation of parameters for classification. Due to the assumption of independence of variables only variances are required for estimation instead of full covariance matrix.

This classifier is best suited when there is high dimensional feature data set. Let C represents a class. The probabilistic model for this class is conditional model i.e. P(C/$X_1, X_2, \ldots\ldots, X_m$) over dependent response variable. From application of Bayes theorem we get:

$$P(C/X_1, X_2, \ldots\ldots, X_m) \alpha\ P(C)\ p(X_1, X_2, \ldots\ldots, X_m/C)$$

The prior probability of j-th class is: $P(C=j) = \frac{Number\ of\ class\ j\ samples}{Total\ numbers\ of\ samples}$

The likelihood function $p(X_1, X_2, \ldots\ldots, X_m)/C)$ can be written as $\prod_{i=1}^{m} p(X_i/C)$ under the assumption of conditional independence of features. Now a new instance can be classified with maximum posterior probability obtained as.

Arg $max_{cj} \in {}_c\ P(C_j)\pi_c\ P(X_i/C_j)$

This technique was further extended by Demichelis et al. (2006).

Logistic Regression: Let Y denotes the levels of sub-functions within each function of the genes. The number of level may varies for each function. This is also known as the response variable, which is nominal in nature i.e. denoting any sub-function with any level does not change the interpretation and analysis technique. Now, the columns of X matrix represent statistics related to each gene which are explanatory variable for a given response i.e. sub-functions of a gene. Let there are K+1 possible response levels, then multinomial logistic regression model can be written as

$$\text{Log}\left[\frac{P_r\ (y=i/X)}{P_r\ (y=K+1/X)}\right] = \beta_{oi} + \beta_i' X_i\ ,\ i=1,2,\ldots..,\ K$$

where, β_{oi} , i=1,2,….,K are intercept parameter and $\beta_1, \beta_2, \ldots\ldots, \beta_K$ are K vectors of Gx1 dimension for slope parameters. This model can be fitted using method of maximum likelihood using either Fisher scoring algorithm or New-Raphson algorithm. The likelihood that g-th gene will have response level y_j can be obtained as

$$P_r(Y=y_j\ /x_j\ ) = \begin{cases} \frac{\text{Exp}\left(\beta_{oi}+x_j^i\beta_i\right)}{1+\sum_{m=1}^{K}\text{Exp}\left(\beta_{om}+\bar{x}_j'\beta_m\right)}, & 1 \le y_j = i \le k \\ \frac{1}{1+\sum_{m=1}^{K}\text{Exp}\left(\beta_{om}+\bar{x}_j'\beta_m\right)}, & y_j = k+1 \end{cases}$$

The model fitting information for the reliability of estimated probability, following criterion may be calculated for j-th observation.

$$-2\log L = -2\sum_j \frac{w_j}{\sigma^2} f_j \log P_r(Y=yj/xj)$$

where, w_j and f_j are weight and frequency of j-th observation and σ^2 is the dispersion parameter. Further, Akaike Information Criterion (AIC) can also be obtained as

$$AIC = -2\log L + 2p$$

where , p is the number of parameters in the model. Cox and Snell (1989) proposed following coefficient of determinant for model fitting information

$$R^2 = 1-\left\{\frac{L(O)}{L(\hat{\beta})}\right\}^{2/n}$$

where L(O) is the likelihood of the interrupt model only and $L(\hat{\beta})$ is the likelihood of the specified model for sample size n. The maximum value of R^2 is $R^2_{max} = 1 - \{L(O)\}^{2/n}$. The adjusted coefficient of determinant (Magelkerke 1991) can be written as

$$R^2_{adj} = \frac{R^2}{R^2_{max}}$$

The values of R^2 and R^2_{adj} are found to be the best criterion for indicating better models while fitting. The functional prediction re-substitutions accuracy has been also estimated from misclassification errors matrices. Further, sensitivity and specificity has also calculated.

Methods for Prediction Error Estimation

It is commonly acknowledged that there is a bias-variance tradeoff in estimating prediction errors. In the conventional *n>p* situation, the .632+ bootstrap is very popular for having low variability and only moderate bias. However, the work of Molinaro *et. al.* (2005) and Jiang and Simon (2007) suggest that the .632+ bootstrap can run into problems in the *n<p* situation. Jiang and Simon (2007) proposed a repeated leave-one-out bootstrap (RLOOB) method and an adjusted bootstrap method. The performance of adjusted bootstrap method is robust in various situations and it achieves a good compromise in the bias-variance tradeoff. The techniques of prediction errors described here are taken from (Ahn and Moon, 2010). In this section, main concentration is given to the bootstrap related methods for estimating prediction errors.

Leave-one-out cross-validation: Cross-validation (Stone (1974)) avoids this problem by removing the data point to be predicted from the learning set. The leave-one-out cross-validation estimate can be expressed as

$$\hat{e}_n^{LOOCV} = 1/n \sum_{i=1}^{n} I\{y_i \neq r(t_i, x_{(-i)})\},$$

where $x_{(-i)}$ represents the learning set with x_i removed . It calculates the rate of misclassified responses when predicting for each specimen using a learning set containing all other observations in the sample. Correct application of the method to high dimensional microarray data requires feature selection for every leave-one-out learning set $x_{(-i)}$ of size $n-1$. The leave-one-out cross-validation produces almost unbiased estimate for the prediction error and has been a common choice for small sample problems. The investigation of Molinaro et al. (2005) suggests that the leave-one-out cross-validation method performs no worse than other cross-validation methods and split sample methods in genomic studies with small to moderate sample sizes. However, when the sample size is small, the leave-one-out cross-validation method is often criticized for having very large variation. The large variability is ascribed mainly to the similarity between the leave-one-out learning sets $x_{(-i)}$, i=1,...n, and the sparseness of the data. The similarity between the sets $x_{(-i)}$ results in large covariance between the terms of $\hat{e}_n^{LOOCV}$, hence increases the overall variance of the estimate.

Ordinary Bootstrap: Ordinary Bootstrap method given by Efron (1998) has the problem that the learning and test sets overlap. In this method bootstrap samples of size n are repeatedly drawn from the original data x by simple random sampling with replacement. In this, a prediction rule is built on a bootstrap sample and tested on the original sample, averaging the misclassification rates across all bootstrap replications gives the ordinary bootstrap estimate. This method seriously underestimates the prediction error since a subset of data is used both in building and in assessing the prediction model.

Bootstrap Cross-Validation: This method is proposed by Fu *et. al.* (2005) to handle small sample problems. The procedure generates B bootstrap samples of size n from the observed sample and then calculates a leave-one-out cross-validation estimate on each bootstrap sample. Averaging the B cross-validation estimates gives the bootstrap cross-validation estimate for the prediction error. The paper of Fu *et. al.* (2005) did not carefully address the issue of feature selection. When the method is applied to high dimensional gene expression data, it is to be noted that feature selection must be conducted in this method on every leave-one-out learning set derived from every bootstrap sample. Since an original observation can appear more than once in a bootstrap sample, a leave-one-out learning set may overlap with the left out item when the cross-validation procedure is applied on a bootstrap sample. Consequently, the bootstrap cross-validation method tends to underestimate the true prediction error.

Leave-One-Out Bootstrap: The leave-one-out bootstrap procedure given by Efron (1983) generates a total of B bootstrap samples of size n. Each observed specimen is predicted repeatedly using the bootstrap samples in which the particular observation does not appear. In this way, the method avoids testing a prediction model on the specimens used for constructing the model. The leave-one-out bootstrap estimate is given by

$$\hat{e}_n^{LOOBS} = 1/n \sum_{i=1}^{n} 1/|C_i| \sum_{b \in C_i} I\{y_i \neq r(t_i, x^{*,b})\}$$

where C_i is the collection of bootstrap samples not containing observation i and $|C_i|$ is the number of such bootstrap samples. Feature selection and class prediction should be performed on each bootstrap sample $x^{*,b}$, $b=1,\ldots,B$.

The leave-one-out bootstrap is basically a smoothed version of the leave-one-out cross-validation. To see this, the bootstrap samples in C_i can be viewed as random samples of size n generated from the leave- i -out data set $x_{(-i)}$. Bootstrap samples are more different between each other than the original leave-one-out sets. Moreover, for each specimen i, the leave-one-out bootstrap method averages on the errors from the multiple predictions made on the bootstrap samples in C_i. As a result, the leave-one-out bootstrap estimate has much smaller variability than the leave-one-out cross validation estimate. On the other hand, a bootstrap sample of size contains roughly .632n distinct observations from the original sample. It is often inadequate to represent the distribution of the original data when the sample size n is small. Hence the leave-one-out bootstrap estimate tends to overestimate the true prediction error.

Out-of-Bag Estimation: The out-of-bag estimation procedure is given by Breiman (1996). The out-of-bag estimate for the prediction error is a by-product of bagging predictors. The out-of-bag estimate is the misclassification rate when predicting for each observation by the class that wins the majority votes from the multiple predictions, made on the bootstrap samples in which the particular observation is out-of-bag (i.e., not included). The out-of-bag estimation makes an interesting comparison to the leave-one-out bootstrap. The out-of-bag estimation employs a majority vote on the multiple predictions made for observation i based on the bootstrap samples in C_i, while the leave-one-out bootstrap takes an average on errors of these predictions. The out-of-bag estimation can be viewed as a non-smooth variant and we envisage it to have larger variability than the leave-one-out bootstrap when the sample size is small.

632+ Bootstrap: The .632+ bootstrap is proposed by Efron and Tibshirani (1997) in order to reduce the upward bias of the leave-one-out bootstrap. The estimate has the form

$$\hat{e}_n^{.632+} = w\hat{e}_n^{LOOBS} + (1-w)\hat{e}_n^{RS}$$

where the weight w is between 0 and 1 and $\hat{e}_n^{RS}$ is the resubstitution estimate. Taking $w = 0.632$ gives the **.632 bootstrap** originally proposed by Efron (1983). When the resubstitution error is zero, the .632 bootstrap estimate becomes $0.632\hat{e}_n^{LOOBS}$, this results in systematic downward bias when there are no class differences. The .632+ bootstrap aims to circumvent this problem by increasing the weight w with respect to the growing level of overfitting. It often performs well in classification problems with $n > p$. For microarray data with $n < p$, the overfitting

problem always exists and the resubstituion error estimate is often close to zero. The .632+ bootstrap tends to put too much weight on the leave-one-out bootstrap estimate in this situation.

Repeated Leave-One-Out Bootstrap (RLOOB) and Adjusted Bootstrap: The Repeated Leave-One-Out Bootstrap and an Adjusted Bootstrap method are proposed by Jiang and Simon (2007). For every original sample x, leave out one observation at a time and denote the resulting sets by $x_{(-1)},\ldots, x_{(-n)}$. From each leave-one-out set $x_{(-i)}$, draw B_1 bootstrap learning sets of size ln. Build a prediction rule on every bootstrap learning set generated from $x_{(-i)}$ and apply the rule on the test observation x_i. The repeated leave-one-out bootstrap estimate is the misclassification rate calculated across all the bootstrap runs and all n observations. It can be expressed as

$$\hat{e}_n^{RLOOB}(l) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{B_1}\sum_{b_i=1}^{B_1} I\{y_i \neq r(t_i, x_{(-i)}^{*,b_i})\}$$

where $x_{(-i)}^{*,b_i}$ is the b_i th bootstrap learning set of size ln drawn from the set $x_{(-i)}$. Feature selection should be carried out on every bootstrap learning set $x_{(-i)}^{*,b_i}$ for $b_i = 1,\ldots, B_i$ and $i= 1,\ldots,n$.

Let $c(l)$ be the chances that an observation appears in a bootstrap sample of size ln. A simple probabilistic argument indicates that $c(l) \approx 1-e^{-l}$. A bootstrap sample of size ln contains approximately $c(l).n$ distinct observations from the original sample. For example, for l =1,2,3, the number of distinct observations is about $0.632n$, $0.865n$, $0.95n$ respectively. With l = 1, the repeated leave-one-out bootstrap closely resembles the leave-one-out bootstrap procedure. As l increases, a bootstrap learning set for a left-out item contains more distinct observations. On one hand, the method acquires additional accuracy and brings a reduction on the upward bias. On the other hand, the bootstrap learning sets obtained from the same leave-one-out set become more similar in structure and this raises the variability of the estimation.

The learning behaviour of the repeated leave-one-out bootstrap can be modelled as a function of the number of distinct observations included in the bootstrap learning sets. The trend of a learning process as a function of sample size is often modelled in the machine learning literature by a flexible curve following an inverse power law. Let m be the expected number of distinct observations to appear in a bootstrap sample of size ln. Let $e(m)$ be the expected error rate given the observed sample using the repeated leave-one-out bootstrap method with bootstrap learning sets of size ln. Ideally, $e(m)$ should follow the inverse power law

$$e(m)= am^{-\alpha} + b$$

where a , α and b are the parameters.

The Adjusted Bootstrap method estimates the prediction error as follows. Pick J bootstrap learning set sizes, $l_j n$, j =1,..., J . Compute the repeated leave-one-out bootstrap estimate $\hat{e}_n^{RLOOB}(l_j)$ with bootstrap learning sets of size $l_j n$. Denote $\hat{e}_n^{RLOOB}(l_j)$ by $e(m_j)$ where $m_j = c(l_j).n$ is the expected number of distinct original observations in a bootstrap learning set of size $l_j n$. Fit an empirical learning curve of the form $e(m_j) = am_j^{-\alpha} + b$ with j=1, …, J. The estimates $\hat{a}$, $\hat{\alpha}$ and $\hat{b}$ for the parameters are obtained by minimizing the non-linear least squares function $\sum_{j=1}^{J}\left\{e(m_j) - am_j^{-\alpha} - b\right\}^2$.

The adjusted bootstrap estimate for the prediction error is given by

$$\hat{e}_n^{ABS} = \hat{a}n^{-\hat{\alpha}} + \hat{b}.$$

It is the fitted value on the learning curve as if all original observations contributed to an individual bootstrap learning set.

In practice, the choice of l can range from somewhere close to 1 to a value greater than 5. Repeated leave-one-out bootstrap estimates typically have lower variability than leave-one-out cross-validation. They have an upward bias that decreases and their variability increases with the expected number of distinct original observations selected in bootstrap. Fitting an inverse power law curve to a series of repeated leave-one-out bootstrap values enables us to define a conservative estimate (not subject to downward bias) that provides a compromise between estimates with large variability and large upward bias. Inverse power law curve is a flexible way to model a learning process, and is quite typical in describing machine learning, human and animal learning behaviour (Shrager et. al. (1988)). Mukherjee et al. (2003) studied sample size requirements in microarray classification using a similar learning curve.

2. Unsupervised Machine learning: Clustering

The most common unsupervised learning is Cluster analysis. It is more primitive technique in that no assumptions are made concerning the number of groups or the group structure. Grouping is done on the basis of similarities or distances (dissimilarities). The inputs required are similarity measures or data from which similarities can be computed.

Similarity measures

Most efforts to produce a rather simple group structure from a complex data set necessarily require a measure of “closeness”, or “similarity”. There is often a great deal of subjectivity involved in the choice of a similarity measure. Important considerations include the nature of the variables (discrete, continuous, binary) or scales of measurement (nominal, ordinal, interval, ratio) and subject matter knowledge. When items (units or cases) are clustered,

proximity is usually indicated by some sort of distance. On the other hand, variables are usually grouped on the basis of correlation coefficients or like measures of association.

Distances and Similarity Coefficients for Pairs of Items

The Euclidean (straight-line) distance between two p-dimensional observations (items)

$x = [x_1, x_2, ..., x_p]'$ and $y = [y_1, y_2, ..., y_p]'$ is

$$d(x,y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + ... + (x_p - y_p)^2} = \sqrt{(x - y)'(x - y)} \quad (1)$$

where as the statistical distance between the same two observations is of the form

$$d(x,y) = \sqrt{(x - y)' A (x - y)} \quad (2)$$

Ordinarily, $A=S^{-1}$, where S contains the sample variances and covariances. However, without prior knowledge of the distinct groups, these sample quantities cannot be computed. For this reason Euclidean distance is often preferred for clustering.

Another distance measure is the Minkowski metric, given as

$$d(x - y) = \left[\sum_{i=1}^{p} |x_i - y_i|^m \right]^{1/m} \quad (3)$$

Genetic Distance

Genetic distance is "that difference between two entities that can be described by allelic variation" (Nei, 1973). This definition was later elaborated by Nei (1987) as "the extent of gene differences…between populations or species that is measured by some numerical quantity". A more comprehensive definition of genetic distance is "any quantitative measure of genetic difference, be it at the sequence level or the allele frequency level that is calculated between individuals, populations or species" (Beaumont et al., 1998).

On the basis of data obtained by measurement of quantitative traits in inbred lines, Smith et al. (1991) suggested a measure of genetic distance as follows:

$$d_{(i,j)} = \sum \left[(T_{1(i)} - T_{2(i)})^2 / \text{var}\, T_{(i)} \right]^{1/2} \quad (4)$$

the var $T_{(i)}$ is the variance for the ith trait over all inbreds.

Various genetic distance measures have been proposed for analysis of molecular marker data for the purpose of genetic diversity analysis. For molecular marker data where the amplification products may be equated to alleles, as in case of simple sequence repeats (SSRs) and restriction fragment length polymorphisms (RFLPs), allele frequencies can be calculated. The genetic distance between individual i and j can be estimated using the formula,

$$d(i,j) = Cons\tan t\left(\sum_{a=1}^{n}\left|X_{ai} - X_{aj}\right|r\right)^{1/r} \quad (5)$$

where X_{ai} is the frequency of the allele a for individual i, n is number of alleles per locus, and r is constant based on the coefficient used. In its simple form (*i.e.*, when r=1), genetic distance can be calculated as

$$d_{1ij} = \tfrac{1}{2}\sum_{a=1}^{n}\left|X_{ai} - X_{aj}\right| \quad (6)$$

when r = 2, d_{ij} is referred to as Rogers' (1972) measure of distance (RD), where

$$RD_{ij} = \tfrac{1}{2}\left[\sum\left(X_{ai} - X_{aj}\right)^2\right]^{1/2} \quad (7)$$

Although allele frequencies can be calculated for some of the molecular markers, the data is most widely employed to generate a binary matrix for statistical analysis. The commonly used measures of genetic distance or genetic similarity (GS) using such binary data are (i) Nei and Li's (1979) coefficient(GD_{NL}), (ii) Jaccard's (1908) coefficient (GD_j), (iii) simple matching coefficient (GD_{SM}) (Sokal and Michener, 1958), and (iv)

Modified Rogers' distance (GD_{MR}). Genetic distances determined by these measures can be estimated as follows:

$$\begin{aligned} GD_{NL} &= 1-\left[2N_{11}/(2N_{11}+N_{10}+N_{01})\right] \\ GD_j &= 1-\left[N_{11}/(N_{11}+N_{10}+N_{01})\right] \\ GD_{SM} &= 1-\left[N_{11}/(N_{00})/(N_{11}+N_{10}+N_{01}+N_{00})\right] \\ GD_{MR} &= \left[(N_{10}+N_{01})/2N\right]^{0.5} \end{aligned} \quad (8)$$

where N_{11} is the number of bands – alleles present in both individuals; N_{00} is number of bands-alleles absent in both individuals; N_{10} is the number of bands-alleles present only in the individual I; N_{01} is the number of bands-alleles present only in the individual j; and N represents the total number of bands-alleles. Appropriate choice of a genetic distance measure, on the basis of the type of the variable and scale of measurement, is an important component in the analysis of genetic diversity among set of genotypes (Mohammadi and Prasanna, 2003).

Clustering Methods

Distance-based clustering methods can be categorized into two groups: hierarchical and nonhierarchical.

Hierarchical Clustering methods

Hierarchical clustering methods proceed by either a series of successive mergers or by a series of successive divisions. *Agglomerative hierarchical methods* start with the individual objects. Thus, there are initially as many clusters as objects. The most similar individuals are first groped, and these initial groups are merged according to their similarities. Eventually, as the similarity decreases, all subgroups are fused into a single cluster. The following are the steps in the agglomerative hierachical clustering algorithm for grouping N objects

i. Start with N clusters, each containing a single entity and an N x N symmetric matrix of distances (or similarities) □□.*ikdD*□

ii. Search the distance matrix for the nearest (most similar) pair of clusters. Let the distance between "most similar" clusters U and V be d_{uv}.

iii. Merge clusters U and V. Label the newly formed cluster (UV). Update the entries in the distance matrix by (a) deleting the rows and columns corresponding to clusters U and V and (b) adding a row and column giving the distances between cluster (UV) and the remaining clusters.

iv. Repeat Steps 2 and 3 a total of N – 1 times. (All objects will be in a single cluster at termination of the algorithm). Record the identity of clusters that are merged and the levels *(distances or similarities) at which the mergers take place.

Among various agglomerative hierarchical methods like, single linkage, complete linkage, average linkage, centroid, Ward's methods, the UPGMA (Unweighted Paired Group Method using Arithmetic averages) (Sneath and Sokal, 1973; Panchen, 1992) is the most commonly adopted clustering algorithm, followed by the Ward's minimum variance method (Ward, 1963). For more details on hierarchical clustering methods reference may be made to Johnson and Wichern (1996).

Non-hierarchical Clustering methods

The nonhierarchical clustering procedures do not involve construction of dendrogams or trees. These procedures, also frequently referred to as "K-means clustering", are based on "sequential threshold", "parallel threshold", or "optimizing" approaches for assigning individuals to specific clusters, once the number of clusters to be formed is specified (Everitt, 1980). MacQueen [17] suggests the term K-means for describing his algorithm that assigns each item to the cluster having the nearest centroid (mean). In its simplest version, the process is composed of these three steps.

i. Partition the items into K initial clusters.
ii. Proceed through the list of items, assigning an item to the cluster whose centroid (mean) is nearest. (Distance is usually computed using Euclidean distance with either standardized or unstandardized observations). Recalculate the centroid for the cluster receiving the new item and for the cluster losing the item.
iii. Repeat Step 2 until no more reassignments take place.

Rather than starting with a partition of all items into K preliminary groups in Step 1, we could specify K initial centroids (seed points) and then proceed to Step 2.

The final assignment of items to clusters will be, to some extent, dependent upon the initial partition or the initial selection of seed points. Experience suggests that most major changes in assignment occur with the first reallocation step. For performing nonhierarchical clustering procedure different statistical packages such as SAS [FASTCLUS] and SPSS [QUICK CLUSTER] are available. Nonhierarchical clustering methods are rarely used for analysis of intraspecific genetic diversity in crop plants. The primary reason could be the lack of prior information about the optimal number of clusters that are required for accurate assignment of individuals.

Choice of a Clustering Method

UPGMA method has been widely used in the past literature. Although some studies indicated the relative advantages of UPGMA clustering algorithm in terms of consistency in grouping biological materials with relationships computed from different types of data, a single clustering method might not be always optimal or effective in revealing genetic associations. Despite some favourable attributes in UPGMA, the underlying assumptions are rarely met. Five clustering methods, namely UPGMA, UPGMC (Unweighted Paired Group Method using Centroids), Single Linkage, Complete Linkage, and Median, were compared for their utility in revealing genotype associations in barley germplasm collections (Peeters and Martinelli, 1989). UPGMA and UPGMC were found to be almost comparable with a relatively high level of accuracy, in accordance with pedigrees, compared to other methods. Single Linkage and

Median clustering methods led to "chaining effect", which gave poor resolution of individual groups and complicated the interpretation of results.

One way of comparing the efficiency of different clustering algorithms is through estimation of the "cophenetic correlation coefficient", which is a product moment correlation coefficient measuring agreement between the dissimilarity-similarity indicated by a phenogram-dendrogram as output of analysis and the distance-similarity matrix as input of cluster analysis. A method yielding a high cophenetic correlation coefficient can be considered as an appropriate method for a particular analysis (Romesburg, 1984). The degree of fit can be interpreted subjectively as; $0.9 \leq r$, very good fit; $0.8 \leq r < 0.9$, good fit; $0.7 \leq r < 0.8$, poor fit; $r < 0.7$, very poor fit (Rohlf, 1992). However, a low cophenetic correlation coefficient does not mean that the dendrogram has no utility, but only indicates that some distortion might have occurred. For a large sample of individuals, the cophenetic correlation coefficients have similar values and are not affected by the number of characters.

Determination of optimal number of clusters

Another important aspect in cluster analysis is determining the optimum number of clusters or number of acceptable clusters. In essence, this involves deciding where to "cut" a dendrogram to find the true or natural groups. An "acceptable cluster" is defined as "a group of two or more genotypes with a within-cluster genetic distance less than the overall mean genetic distance and between cluster distances greater than their within cluster distance of the two clusters involved" (Brown-Guedira *et al*., 2000). Thompson *et al*., (1998) find that consistency of clustering based on different methods of grouping, provides strong evidence that natural clusters are present and they also used the multidimensional scaling for the evidence of the major grouping of genotypes in the cluster analysis. Also suggested that the second eigen value of similarity matrix to set at 0.75 to be certain that most of the variation is explained by the first PC. Using molecular marker data, Melchinger (1993) compared PCA, Principal Coordinate analysis (PcoA) and cluster analysis with respect to their efficiency in analyzing genetic diversity in crop plants. Messmer *et al*., (1992) suggested that to extract maximum information for molecular marker data, PCA or PCoA could be used in combination with cluster analysis.

References

Ahn, H. and Moon H. (2010). Statistical Bioinformatics edited by Lee, J.K, Wiley and Blackwell Publications, New Jersey.

Barrett, B.A., and K.K. Kidwell. 1998. AFLP-based genetic diversity assessment among wheat cultivars from the Pacific Northwest. *Crop Sci.* 38:1261–1271.

Beaumont, M.A., K.M. Ibrahim, P. Boursot, and M.W. Bruford. 1998. Measuring genetic distance. p. 315–325. *In* A. Karp et al. (ed.) Molecular tools for screening biodiversity. Chapman and Hall, London.

Breiman, L., (2001). Random Forest. Machine Learning, 45: 5-32

Breiman, L., Freidman, J.H., Olshen, R.A., and Stone, C.J. (1984). Classification and Regression Tree Belmont, CA: Wadsworth.

Brown-Guedira, G.L., J.A. Thompson, R.L. Nelson, and M.L. Warburton. 2000. Evaluation of genetic diversity of soybean introductions and North American ancestors using RAPD and SSR markers. *Crop Sci*. 40:815–823.

Dudoit, S., Fridlyand, J., and Speed T.P. (2002) Comparison of discriminant method for classification of tumors using gene expression data. J. Am. Stat. Assoc., 97:77-87.

Everitt, B. 1980. Cluster analysis. 2nd edition, Halstead Press, New York.

Excoffier, L., P. Smouse, and J. Quattro. 1992. Analysis of molecular variance inferred for metric distances among DNA haplotypes: application to human mitochondrial DNA restriction data. *Genetics* 131:479–491.

Friedman, J. (1997). On bias, variance, 0/1-loss, and the curse-of-dimensionality. Data Mining Knowledge Discov.1, 55-77.

Huff, D.R., R.Peakall, and P.E. Smouse. 1993. RAPD variation within and among natural populations of outcrossing buffalograss [*Buchloe dactyloides* (Nutt.) Engelm.]. *Theor. Appl. Genet.*, 86:927–34.

Jaccard, P. 1908. Nouvelles researches sur la distribution florale. *Bull. Soc. Vaudoise Sci. Natl.* 44:223–270.

Johns, M.A., P.W. Skrotch, J. Neinhuis, P. Hinrichsen, G. Bascur, and C. Munoz-Schick. 1997. Gene pool classification of common bean landraces from Chile based on RAPD and morphological data. *Crop Sci.* 37:605–613.

Johnson, A.R., and D.W. Wichern. 1996. Applied multivariate statistical analysis. Prentice-Hall, Englewood Cliffs, NJ.

Tibshirani, R., Hastie, T., Narasimhan, B., and Chu, G. (2002). Diagnosis of multiple cancer types by shrunken centriods of gene expression. Proc. Natl. Acad.Sci. U.S.A, 99: 6567-6572

Vapnik, V. (1995). The nature of statistical learning theory, New York, Springer.

Supervised Machine Learning (Practical)

setpath:setwd("path of working directory");

 # Reading entire data set into matrix named with 'bcdata'

bcdata<- read.csv("Filename".txt")

 # Data segregation into dependent and independent variables

databcall <- subset(bcdata,select=c(-Samplecodenumber,-Class))
classesbcall <- subset(bcdata,select=Class)

 # Data partitioning into test and training sets

databctrain <- databcall[1:400,]
classesbctrain <- classesbcall[1:400,]
databctest <- databcall[401:699,]
classesbctest <- classesbcall[401:699,]

 # SMV modeling

 # Install the packages

install.packages("e1071")
model <- svm(databctrain, classesbctrain)
pred <- predict(model, databctest)
print(model)
summary(model)
# tune(svm, train.x=databctrain, train.y=classesbctrain, validation.x=databctest, validation.y=classesbctest, ranges = list(gamma = 2^(-1:1), cost = 2^(2:4)), control = tune.control(sampling = "fix"))

 # Confusion Matrix

table(pred,t(classesbctest))

 # Random Forest

```
Library(randomForest)
model_rf <-randomForest(V1~., data=databctrain, ntree=1000, proximity=TRUE)
rf_Predt<-predict(model_rf, databctest)
```

 # Artificial Neural Network

 # Install the packages

```
install.packages("neuralnet")
install.packages("nnet")
install.packages("ggplot2")
install.packages("dplyr")
install.packages("reshape2")
```

 # Load the libraries

```
library(neuralnet)
library(nnet)
library(ggplot2)
library(dplyr)
library(reshape2)
```

 # Load the data

```
data("iris")
```

 # Set the seed

```
set.seed(123)
```

 # Distribution of each feature in the iris data set

```
exp_iris <- melt(iris)
exp_iris %>%
  ggplot(aes(x = factor(variable), y = value)) +
  geom_violin() +
  geom_jitter(height = 0, width = 0.1, aes(colour = Species), alpha = 0.7) +
  theme_minimal()
```

```
# Convert observation class and Species into vector.

labels <- class.ind(as.factor(iris$Species))

# Generic function to standardize a column of data.

stand <- function(x){
  (x-min(x))/(max(x)-min(x))
}

# Standardize the data

iris[, 1:4] <- lapply(iris[, 1:4], stand)

# Combine level and predictors

combine_iris <- cbind(iris[,1:4], labels)

# Formula for fitting

f <- as.formula("setosa + versicolor + virginica ~ Sepal.Length + Sepal.Width + Petal.Length
+ Petal.Width")

# Fitting of the data
iris_fit <- neuralnet(f, data = combine_iris, hidden = c(16, 12), act.fct = "tanh", linear.output
= FALSE)

#Plot Neural Network

plot(iris_fit)

# Accuracy of Neural Network

iris_preds <- neuralnet::compute(iris_fit, combine_iris[, 1:4])
original_vals <- max.col(combine_iris[, 5:7])
pr.nn <- max.col(iris_preds$net.result)
print(paste("Model Accuracy: ", round(mean(pr.nn==original_vals)*100, 2), "%.", sep = ""))
```

R code for Un-Supervise learning
(k-mean Clustering and Hierarchical Clustering)

```
#setwd("F:\\Iris demo")

library(datasets)
data(iris)
dim(iris)

############## K-means Clustering ###############
iris.new<- iris[,c(1,2,3,4)]
iris.class<- iris[,"Species"]
result<- kmeans(iris.new,3)
result$size
result$centers
result$cluster
par(mfrow=c(2,2), mar=c(5,4,2,2))
plot(iris.new[c(1,2)], col=result$cluster)
plot(iris.new[c(1,2)], col=iris.class)
plot(iris.new[c(3,4)], col=result$cluster)
plot(iris.new[c(3,4)], col=iris.class)
table(result$cluster,iris.class)

############ Heirarchical Clustering ############
m <- hclust(dist(iris[,1:4]), method="ave")
plot(m, cex=0.5)
clusters = cutree(m, 3)
table(clusters, iris$Species)
```

Genome Editing Using CRISPR/Cas9

(Practical)

1. Genome Editing

Genome editing is a way of making specific changes to the DNA of a cell or organism. An enzyme cuts the DNA at a specific sequence, and when this is repaired by the cell a change or 'edit' is made to the sequence.

- Genome editing is a technique used to precisely and efficiently modify DNA within a cell
- It involves making cuts at specific DNA sequences with enzymes called 'engineered nucleases'.
- Genome editing can be used to add, remove, or alter DNA in the genome
- By editing the genome the characteristics of a cell or an organism can be changed.

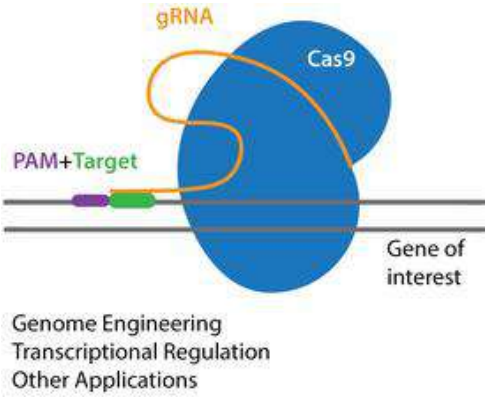
2. Where is genome editing used?

Genome editing can be used:

- **For research:** Genome editing can be used to change the DNA in cells or organisms to understand their biology and how they work.
- **To treat disease:** Genome editing has been used to modify human blood cells that are then put back into the body to treat conditions including leukemia and AIDS. It could also potentially be used to treat other infections (such as MRSA) and simple genetic conditions (such as muscular dystrophy and hemophilia).
- **For biotechnology:** Genome editing has been used in agriculture to genetically modify crops to improve their yields and resistance to disease and drought, as well as to genetically modify cattle that don't have horns.

3. Crispr/Cas9 System

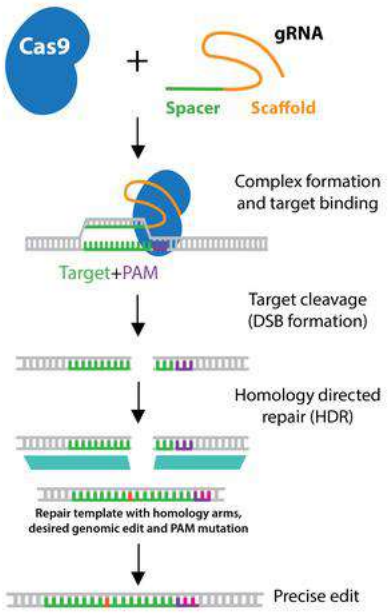
Class 2 Clustered Regularly Interspaced Short Palindromic Repeat (CRISPR) systems, which form an adaptive immune system in bacteria, have been modified for genome engineering. Prior to CRISPR, genome engineering approaches like zinc finger nucleases (ZFNs) or transcription-activator-like effector nucleases (TALENs) required scientists to design and generate a new nuclease pair for every genomic target. Due to its comparative simplicity and adaptability, CRISPR has rapidly become the most popular genome engineering approach.



Engineered CRISPR systems contain two components: a **guide RNA (gRNA or sgRNA)** and a CRISPR-associated endonuclease (Cas protein). The gRNA is a short synthetic RNA composed of a **scaffold** sequence necessary for Cas-binding and a user-defined ~20 nucleotide **spacer** that defines the genomic target to be modified. Thus, one can change the genomic target of the Cas protein by simply changing the target sequence present in the gRNA.

CRISPR was originally employed to knock out target genes in various cell types and organisms, but modifications to various Cas enzymes have extended CRISPR to selectively activate/repress target genes, purify specific regions of DNA, image DNA in live cells, and precisely edit DNA and RNA. Furthermore, the ease of generating gRNAs makes CRISPR one of the most scalable genome editing technologies. This advantage makes CRISPR perfect for genome-wide screens.

FOR FURTHER INFORMATION ON CRISPR PLEASE VISIT: https://www.addgene.org/crispr/guide/



For using CRISPR/Cas9 for gene editing, it is very important to consider the guide RNA sequence and for this bioinformatics is an efficient way for guide RNA, on target and off target predictions.

In this lecture, let's take an example of the **tool CCTop** that essentially gives the desired predictions.

4. CCTop

CCTop provides an intuitive user interface with reasonable default parameters that can easily be tuned by the user. From a given query sequence, CCTop identifies and ranks all candidate sgRNA target sites according to their off-target quality and displays full documentation. CCTop was experimentally validated for gene inactivation, non-homologous end-joining as well as homology directed repair. Thus, CCTop provides the bench biologist with a tool for the rapid and efficient identification of high quality target sites.

REFER TO https://www.ncbi.nlm.nih.gov/pubmed/25909470 FOR DETAILED INFORMATION ON CCTop

Search Guidelines

- **name**: provide a descriptive name for your search. This name will be used for the output files to download so that you can keep track of different searches.
- **Select either single query or batch search**: in the first case, the sequence can be up to 500 nucleotides long. Paste a plain sequence, not fasta format. Only characters representing valid nucleotides (A,a,C,c,G,g,T,t,N,n) will be considered and any other character will be discarded. In the second case you can provide a (multi-)fasta file with any number of sequences and a total size of up to 500KB.
- **PAM type**: apart from the protospacer adjacent motif (PAM) recognized by the Cas9 protein of Streptococcus pyogenes (SP), other motifs have been identified for different bacterial species. In addition, it has been shown that the Cas9 endonuclease in SP can also cleave sgRNA target sites followed by 'NAG', however with efficiency reduced to ~20% , rendering a PAM motif 'NRG'. Selecting a PAM motif other than the one recognized by the Cas9 of SP will disable the use of the core parameters for off-target site search. They have also included the PAM of the Cpf1 endonuclease , from *Acidaminococcus* or *Lachnospiraceae*, that recognizes a 'TTTN' motif.

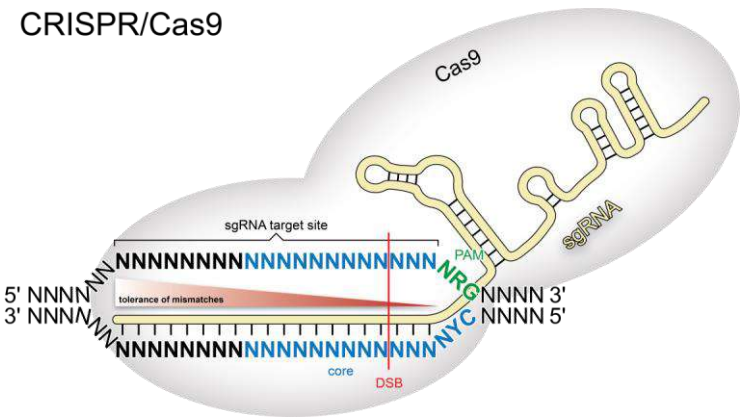
Target selection

- **target site length**: the length of the sgRNA target site, excluding the PAM sequence, can be from 15 to 23 bases.

- **target site 5' limitation**: typical in vitro transcription promoters, like the T7 or the U6, require one or two leading 'G' respectively. However, we and others found that adding one or two Gs to the desired target sequence can also give valid sgRNAs that induce double strand breaks (DSBs). This release of restriction identifies many more target sites in the query sequence (hence 'NN' is the default value).
- **target site 3' limitation**: for C. elegans it has been shown that 'GG' at the 3' site enhances the introduction of a DSB. However, as with the previous field, you can specify any valid sequence of two bases to restrict the list of possible CRISPR/Cas9 target sites.
- **in vitro transcription**: Select your in vitro transcription method. They will provide the forward and reverse oligos to clone each candidate sgRNA into the appropriate vector. If you select "Custom" you can specify your overhangs, which will be appended to the 5' of the repective primers.

Off-target prediction

- **max. total mismatches**: can be set to 0-5. Note: more than four mismatches in total prevent DSB induction
- **max. core length**: Mismatches at a distance to the PAM, will still allow the introduction of a DSB, while mismatches close to the PAM will abolish the introduction of a DSB. The core is a simplified parameter to account for these findings and is defined as the nucleotides adjacent to the PAM (12 by default, can be set to 2-20). The checkbox in front allows to enable/disable the core parameters. The prediction of off-target sites taking into account the core is only available when using the PAM motif from S. pyogenes. In the other cases there is not yet experimental evidence suggesting that the core can be beneficial.
- **max. core mismatches**: can be set to 0-2. Note: in the core, more than 2 mismatches abolishes DSB introduction at the potential off-target position



- **species**: define in which genomic context off-targets should be predicted.

Output

Single query

During search, a page will auto refresh indicating the progress, i.e. how many candidate sgRNAs were identified and which is the current candidate under analysis. After the search process you will automatically be forwarded to the results page. In the following example a genomic sequence from medaka Pax-6 was used.

CCTop - CRISPR/Cas9 target online predictor

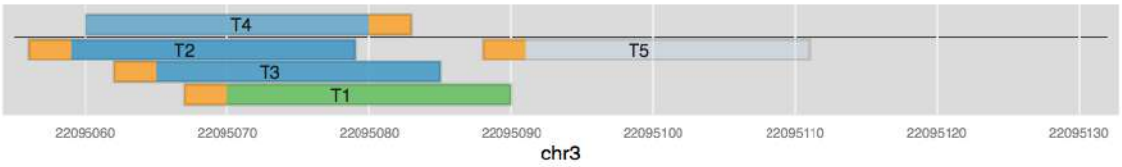
Results for PAX-6

Download full results file here.
Download sgRNAs target sites as fasta file here.
Visualize sgRNA target sites in the UCSC genome browser.

Detailed results

Species: Medaka (Oryzias latipes oryLat2)
Input: ACCACAACCAAGCCACGTGGGAGTCTGGTGTAGCCTCAATGATGCAGAACAGTAAGTTGA TTTCATTTGTGTTATGC

PAM: NGG
Target site length: 20
Target site 5' limitation: NN
Target site 3' limitation: NN
Core length: 12
Core MM: 2
Total MM: 4



Legend for off-target site positon: E = exonic; I = intronic; - = intergenic

T1 out of 5
<Previous Next>
Sequence: GGCTACACCAGACTCCCACGTGG
Oligo pair fwd: TAGGCTACACCAGACTCCCACG rev: AAACCGTGGGAGTCTGGTGTAG

Coordinates	strand	MM	target_seq	PAM	distance		gene name	gene id
chr3:22095068-22095090	-	0	GGCTACAC[CAGACTCCCACG]	TGG	0	E	PAX-6	ENSORLG00000009913
chr2:7142306-7142328	-	4	GGTTCCCC[CAGACACCCACG]	CGG	0	E	NA	XLOC_012239
chr24:5763510-5763532	+	4	GGTGACAT[CAGACTGCCACG]	TGG	0	E	tagapb	ENSORLG00000012139
chr12:1547254-1547276	+	4	GACTACGC[CTGACTCCAACG]	TGG	903	I	ZMIZ2	XLOC_003751

The page provides links:

- to download the full results file (tab separated) like shown in the table at the bottom.
- to download a fasta file containing all identified sgRNA target sites
- to visualize the query sequence in the UCSC browser with color coded sgRNA target site location (this link only appears if the query sequence was of the same origin as the targeted genome and the genome is available through UCSC)

Further, the input parameters are displayed for overview. A graphical representation of the query sequence with the identified sgRNA target sites (colored by score, see below) as well as a full list of all candidates is given ranked by taking into account the number of total off-target sites, the distribution of mismatches and the proximity to exons. It is possible to click on any of the displayed target sites to focus the list below on the output corresponding to this site. For each sgRNA target site, cloning oligonucleotides (5'-3' orientation) are provided depending of the in vitro transcription method selected. For the T7 promoter, if the candidate sgRNA sequence does not start with two Gs, the sequence is extended or the initial bases are changed to obtain the required two Gs at the 5' end. The **extended** or **substituted** bases are given in small case for recognition. For the U6 promoter the same procedure is taken only that in this case only one G at the 5' end is required. In other case, the given overhang sequence is appended to the identified sgRNA target sequence. Detailed information is provided for each potential off-target site (only at most 20 are shown, for the full list refer to the .xls file):

- **genomic coordinates**: with UCSC link, if applicable
- **strand**: orientation of the (off-) target site
- **MM**: number of mismatches
- **target_seq**: off-target sequence with highlighted mismatches in red, core in square brackets
- **PAM**: endogenous PAM of the (off-) target site
- **distance**: distance to the closest exon (0 if target site and exon coordinates overlap; NA for target sites farther than 100kb to the next exon). Further information on the location of the off-target site is provided by a colour code: green = intergenic; yellow = intronic; red = exonic.
- **gene name**: the corresponding gene name
- **gene id**: the corresponding gene id (with ENSEMBL link, if applicable) and identifier. For medaka, additional genes were included based on RNA-seq data from different embryonic stages (unpublished data; XLOC identifier).

The tab separated output file has the same structure as the displayed results table:

	A	B	C	D	E	F	G	H	I	J	K	L
1	Species:	Medaka (Oryzias latipes oryLat2)										
2	Input:	ACCACAACCAAGCCACGTGGGAGTCTGGTGTAGCCTCAATGATGCAGAACAGTAAGTTGATTTCATTTGTGTTATGC										
3	PAM:	NGG										
4	Target site length	20										
5	Target site 5' limit	NN										
6	Target site 3' limit	NN										
7	Core length:	12										
8	Core MM:	2										
9	Total MM:	4										
10												
11	T1	GGCTACACC	1000									
12	Oligo fwd	TAGGCTACACCAGACTCCCACG										
13	Oligo rev	AAACCGTGGGAGTCTGGTGTAG										
14	Chromosome	start	end	strand	MM	target_seq	PAM	alignment	distance	position	gene name	gene id
15	chr3	22095068	22095090	-	0	GGCTACACC	TGG	||||||||[|||||||||||]PAM	0	Exonic	PAX-6	ENSORLG00000009913
16	chr2	7142306	7142328	-	4	GGTTCCCCC	CGG	||-|-|-[[|||||-||||||]PAM	0	Exonic	NA	XLOC_012239
17	chr24	5763510	5763532	+	4	GGTGACATC	TGG	||--|||-[||||||-|||||]PAM	0	Exonic	tagapb	ENSORLG00000012139
18	chr12	1547254	1547276	+	4	GACTACGCC	TGG	|-||||-|[[|-||||||-||||]PAM	903	Intronic	ZMIZ2	XLOC_003751

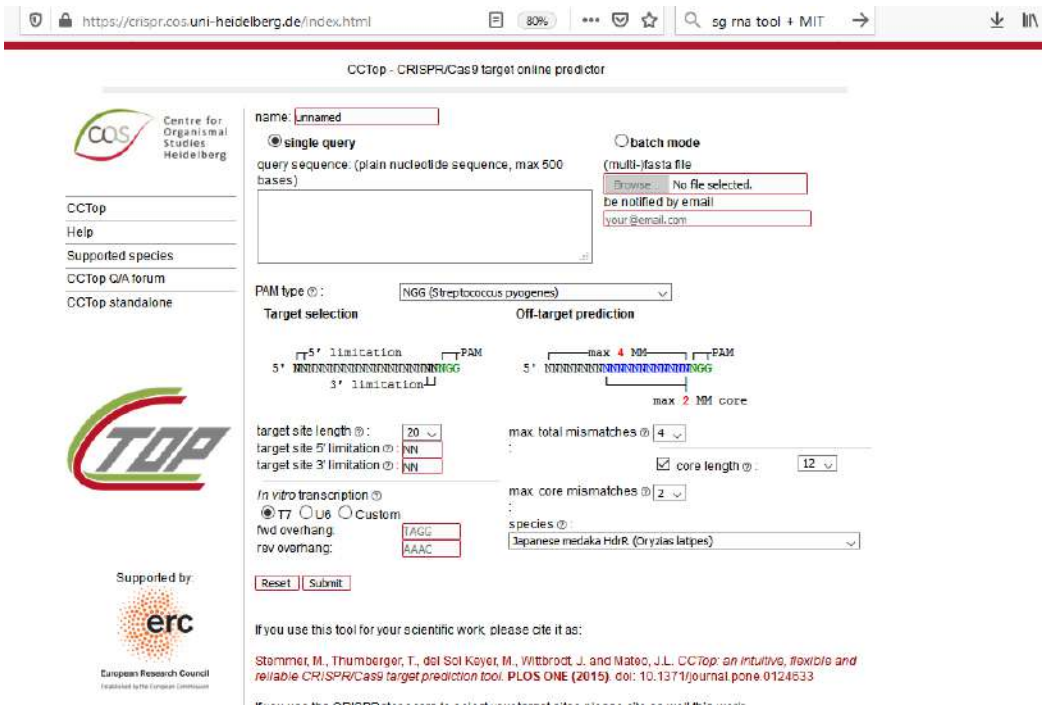
Instead of the color mismatch representation in the target_seq, an additional column is added displaying the alignment of each off-target to the corresponding on-target. Each candidate (target) is labelled with the prefix "T" and a correlative number from 1 to the number of candidates. Candidates are scored from 1000 - suggested best choice to 0 - worst choice. This score takes into account the number of off-targets in the genome, their quality, i.e. number of mismatches and position with respect to the PAM, and the distance to gene exons. The off-target sites for each target site are internally ranked by decreasing likelihood of potential Cas9 activity. If the query sequence is derived from the same genome against which the off-target sites were predicted, the first hit of each target is the candidate target itself, displaying its properties.

Batch query

The result page for a batch search is slightly different. During the search process the page indicates which sequence is being analysed and the refresh time is longer. For this kind of tasks it is advisable to provide a valid email address so that when the search is finished this will be notified with a message to that address. Once the search process is finished successfully the result page will offer a link to download the full set of results in an archive with zip format. Also a link to the specific result page, as described for a single query, will be given for each one of the sequences contained in the input fasta file. The content of the zip archive consists of the bed, fasta and xls file described above and a html file to visualize locally the results in a web browser. Note that for batch searches only a maximum of 50 off-target sites will be considered, if you need the exahustive list of off-target sites you can later run a searach in single sequence mode with the target site of interest.

5. Work Yourself on CCTop

1. Visit: https://crispr.cos.uni-heidelberg.de/index.html



2. Add a query sequence
   For your convenience, let's take the following example:

```
>JF909299.1 Homo sapiens insulin (INS) mRNA, partial cds
CTGGGGACCTGACCCAGCCGCAGCCTTTGTGAACCAACACCTGTGCGGCTCACACCTGGTGGAAGCTCTC
TACCTAGTGTGCGGGGAACGAGGCTTCTTCTACACACCCAAGACCCGCCGGGAGGCAGAGGACCTGCAGG
TGGGGCAGGTGGAGCTGGGCGGGGGCCCTGGTGCAGGCAGCCTGCAGCCCTTGGCCCTGGAGGGGTCCCT
GCAGAAGCGTGGCATTGTGGAACAATGCTGTACCAGCATCTGCTCCCTCTACCAGCTGGAGAACTACTGC
AACTA
```

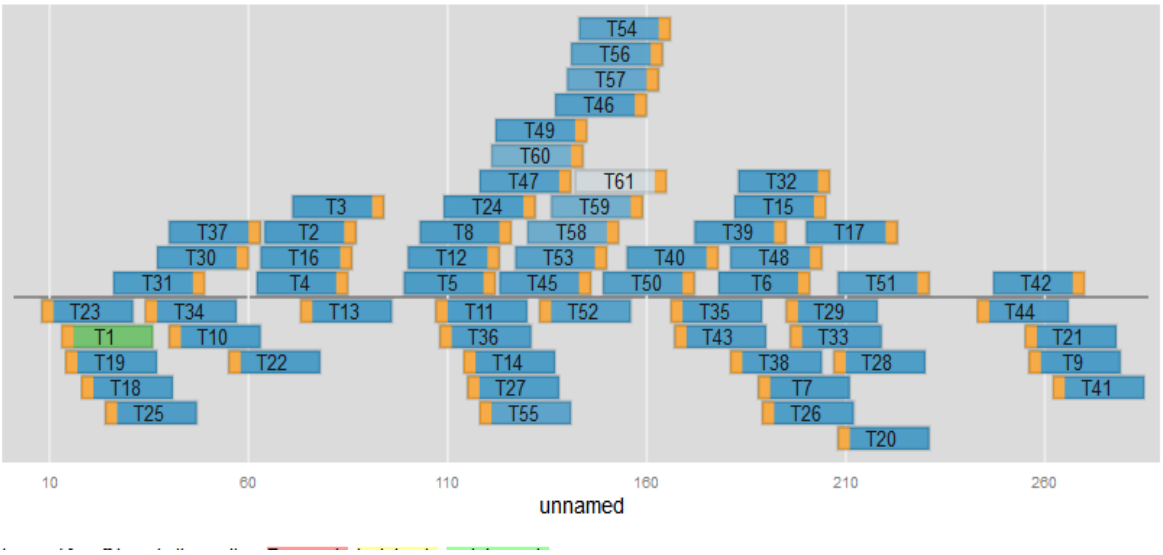
Results for unnamed

Download full results file here.
Download sgRNAs target sites as fasta file here.

Detailed results

Species: Japanese medaka HdrR (Oryzias latipes)
Input: CTGGGGACCTGACCCAGCCGCAGCCTTTGTGAACCAACACCTGTGCGGCTCACACCTGGT GGAAGCTCTCTACCTAGTGTGCGGGGAACGAGGCTTCTTCTACACACCCAAGACCCGCCG GGAGGCAGAGGACCTGCAGGTGGGGCAGGTGGAGCTGGGCGGGGGCCCTGGTGCAGGCAG CCTGCAGCCCTTGGCCCTGGAGGGGTCCCTGCAGAAGCGTGGCATTGTGGAACAATGCTG TACCAGCATCTGCTCCCTCTACCAGCTGGAGAACTACTGCAACTA

PAM:	NGG
Target site length:	20
Target site 5' limitation:	NN
Target site 3' limitation:	NN
Core length:	12
Core MM:	2
Total MM:	4



Legend for off-target site positon: E = exonic; I = intronic; - = intergenic
Legend for the CRISPRater score: LOW efficacy (score<0.56); MEDIUM efficacy (0.56<=score<=0.74); HIGH efficacy (score>0.74)

T1 out of 61

<Previous Next>

Sequence: GGTTCACAAAGGCTGCGGCTGGG

Efficacy score by CRISPRater: 0.63 MEDIUM

Oligo pair fwd: TAGGTTCACAAAGGCTGCGGCT rev: AAACAGCCGCAGCCTTTGTGAA

Coordinates	strand	MM	target_seq	PAM	distance		gene name	gene id
12:29921250-29921272	-	2	GTTTCACA[AAGGCTGCTGCT]	GGG	0	E		ENSORLG00000030536
7:7517528-7517550	-	4	GTTTGACA[AAGGCTCCTGCT]	TGG	1339	-	actr5	ENSORLG00000003806
2:22797532-22797554	-	4	GTTTCAGA[AAGGCAGCGTCT]	CGG	1533	-		ENSORLG00000025753
3:34777496-34777518	+	4	GGATCATA[AAGGCTTCGGGT]	TGG	27769	-	dhcr7	ENSORLG00000015429
3:19551121-19551143	+	4	GGTAAACA[AAGGCTGCAACT]	AGG	17433	-	lmo1	ENSORLG00000006857
4:8335206-8335228	+	4	TGTACACA[AAGGCTGCTGCA]	GGG	1566	I	fbn2b	ENSORLG00000005344
8:19696785-19696807	+	4	GGCTAACA[AAGGCTGCTGCC]	TGG	817	I	cdc42ep1b	ENSORLG00000027921

T2 out of 61

<Previous Next>

Sequence: AGCTCTCTACCTAGTGTGCGGGG

Efficacy score by CRISPRater: 0.54 LOW

Oligo pair with 5' extension fwd: TAggAGCTCTCTACCTAGTGTGCG rev: AAACCGCACACTAGGTAGAGAGCT

Oligo pair with 5' substitution fwd: TAggCTCTCTACCTAGTGTGCG rev: AAACCGCACACTAGGTAGAGAG

Coordinates	strand	MM	target_seq	PAM	distance		gene name	gene id
18:22408336-22408358	+	4	AGTTCACT[ACCCAGTGTTCG]	AGG	4433	-	si:dkey-46i9.1	ENSORLG00000027604

Genome Wide Association Study (GWAS)-Statistical View Point

A genome-wide association study is an approach that involves rapidly scanning markers across the complete sets of DNA, or genomes, of many subjects to find genetic variations associated with a particular trait. Once new genetic associations are identified, researchers can use the information to develop better strategies to detect and manage the trait. Such studies are particularly useful in finding genetic variations that contribute to common, complex traits. In other words, Genome Wide Association Studies (GWAS) is based on correlations between genetic markers (usually Single Nucleotide Polymorphisms, short SNPs) and any measurable trait in a population of individuals. The main motivation in identifications of these associations is to find out new candidates for causal variants in genes (or their regulatory elements) that play a role for the phenotype of interest. This may eventually lead to a better understanding of the genetic components of the trait. Current GWAS usually include the following steps:

- Genotype calling from the raw chip-data and basic quality control.
- Principle Component Analysis (PCA) to detect and possibly correct for population stratification.
- Genotype imputation (using linkage disequilibrium information from HapMap).
- Testing for association between a single SNP and continuous or categorical phenotypes.
- Global significance analysis and correction for multiple testing.
- Data presentation (e.g. using quantile-quantile and Manhattan plots).
- Cross-replication and meta-analysis for integration of association data from multiple studies.

It has been found that (meta-) studies with many thousands (and even ten-thousands) of samples could at best identify a few (dozen) candidate loci with highly significant associations. Although, these unknown associations have been replicated in independent studies, each locus explains but a tiny (<1%) fraction of the genetic variance of the phenotype. Number of reasons could be attributed to this fact. Some important reasons are as follows:

- Estimation of heritability of trait from one generation to another is a problem especially for low heritable traits.
- Often genotype information is incomplete. For example, most analyses used microarrays probing of fractions of SNPs, while many of these SNPs can be imputed accurately using information on linkage disequilibrium. There still remains a significant fraction of SNPs which are poorly tagged by the measured SNPs. Furthermore, rare

variants with a Minor Allele Frequency (MAF) of less than 1% are not accessed at all with SNP-chips, which may nevertheless be the causal agents for many phenotypes. Finally, other genetic variants like Copy Number Variations (CNVs) (or even epigenetics) may also play an important role.

- Current analyses usually only employ additive models considering one SNP at a time with few co-variables and principle components reflecting population sub-structures. This obviously covers a small set of all possible interactions between genetic variants and the environment. Even more challenging task is taking into account purely genetic interactions, since already the number of all possible pair-wise interactions scales like the number of genetic markers squared.

Micro satellites markers are generally used for finding association with a candidate gene or linked region of a chromosomes. This is due to the fact that linkage exists over a very broad region and entire chromosome can be divided only 400-800 DNA markers regions. This can be used for population/family based designs. Using SNPs are more appropriate in other cases but cost plays an important role in this case.

Single Nucleotide Polymorphisms

The modern unit of genetic variation is the Single Nucleotide Polymorphism or SNP. SNPs are single base-pair changes in the DNA sequence that occur with high frequency in a genome. For the purposes of genetic studies, SNPs are typically used as markers of a genomic region, with the large majority of them having hardly any impact on biological systems. SNPs can have functional consequences, however, causing amino acid changes, changes to mRNA transcript stability, and changes to transcription factor binding affinity. SNPs are by far the most abundant form of genetic variation in living organism. SNPs typically have two alleles, meaning within a population there are two commonly occurring base-pair possibilities for a SNP location. The frequency of a SNP is given in terms of the minor allele frequency or the frequency of the less common allele. Rare SNPs i.e. SNPs with low frequency in the population are sometimes referred to as mutations though they can be structurally equivalent SNPs - single base-pair changes in the DNA sequence. In the genetics literature, the term SNP is generally applied common single base-pair changes, and the term mutation is applied to rare genetic variants. In is well known fact that common traits are likely to be influenced by genetic variation that is also common in the population. Also, if common genetic variants influence the trait, the effect size for any one variant must be small relative to that found in rare trait. Therefore, the allele frequency and the population prevalence are completely correlated. However, if a SNP caused a small change in gene expression that has small effect, then the influential allele would be only slightly correlated. Further, if common alleles have small genetic effects, but show heritability then multiple common alleles must influence disease susceptibility. These points suggest that traditional family-based genetic studies are not likely to be successful for complex

traits, prompting a shift toward population-based studies. The frequency with which an allele occurs in the population and the risk incurred by that allele for complex trait are key components to consider when planning a genetic study along with impact of the technology needed to gather genetic information and the sample size needed to discover statistically significant genetic effects. Under these circumstances we need to go for GWAS. GWAS needs large sample sizes and a large panel of genetic markers technology to gather genetic information to discover statistically significant genetic effects.

Linkage disequilibrium (LD) mapping of QTL exploits population level associations between markers and QTL. These associations arise because there are small segments of chromosome in the current population which are descended from the same common ancestor. These chromosome segments, which trace back to the same common ancestor without intervening recombination, will carry identical marker alleles or marker haplotypes, and if there is a QTL somewhere within the chromosome segment, they will also carry identical QTL alleles. There are number of QTL mapping strategies which exploit LD, the simplest of these is the genome wide association test using single marker regression

Genome Wide Association Study

It refers to a method / methodology for interrogating large number of variable points across a genome. As these variations are inherited in groups, or blocks, not all points have to be tested. It is an approach which involves rapidly scanning markers across the complete sets of DNA, or genomes of number of subjects to find genetic variations associated with a particular trait. Once new genetic associations are identified, researchers can use this information to develop better management strategies. Genome-wide association studies were made possible by the availability of chip based microarray technology for assaying one million or more SNPs. Two primary platforms have been used for most GWAS i.e. Illumina and Affymetrix. The Affymetrix platform prints short DNA sequences as a spot on the chip that recognizes a specific SNP allele. Alleles (i.e. nucleotides) are detected by differential hybridization of the sample DNA. Illumina on the other hand uses a bead-based technology with slightly longer DNA sequences to detect alleles. The Illumina chips are more expensive to make but provide better specificity. It is important to note that the technology for measuring genomic variation changing rapidly. Chip-based genotyping platforms are being replaced over the years with inexpensive new technologies for sequencing the entire genome i.e. next-generation sequencing methods. There are two primary classes of phenotypes categorical i.e. binary (case/control) and quantitative. Statistically, quantitative traits are preferred because as they improve power to detect a genetic effect, and often have a more interpretable outcome. The study design for this genetic association differs based on (i) scale of study i.e. genome wide based or genomics based, (ii) marker design, which depends on selection of best marker i.e. microsatellite, SNP and CNV (iii) subject design i.e. based on candidate gene or genome wide screening approach.

The genome wide studies mainly classified in to three categories i.e. cohort studies, family based study and case-control studies. In case of cohort studies, the subjects are assumed to be representative of the population. The phenotypes are used to ascertain the similarity among these subjects irrespective of genetic variations. This technique directly measures the risk and also less biased then case control studies. But it requires long follow up with large sample size. It also very expensive and poorly suited for rare traits. In case of family based studies, the basic assumption is that families are representative of the population of interest and both parents are from same genetic background. The major advantage of this technique is that, it checks for Mendelian in heritance and less prone to spurious associations. In this case, parent phenotypes are not required. It also allows for investigation imprinting and simple logistics techniques are applicable to detect the association. It is cost inefficient, with low power and very sensitive to genotyping errors. Third types of studies are known as case-control studies. In these studies, subjects are drawn from same population and cases represent all cases of the population. These are simple, cheap, and we can use large number of case and control variables. These are optimal for studying rare traits. In this, results are prone to population stratification criterion. In this, batch effect and other biases plays a major role. Generally it gives over estimation for common traits. Mostly, GWAS are used in diseased studies. In case of disease studies, there are three types of diseases as follows:

- Monogenic diseases: This is also a single gene produce disease. Often these disease are severe and appear early in life cycle. For the population as a whole, they are relatively rare. In a sense, these are pure genetic diseases. They do not require any environmental factors to elicit them. Although, nutrition is not involved in the causation of monogenic diseases, these diseases can have implications for nutrition. They reveal the effects of particular proteins or enzymes that also are influenced by nutritional factors.
- Oligogenic diseases: These are conditions produced by the combination of two, three, or four defective genes. Often a defect in one gene is not enough to elicit a full-blown disease, but when it occurs in the presence of other moderate defects, a disease becomes clinically manifest. It is the expectation of human geneticists that many chronic diseases can be explained by the combination of defects in a few (major) genes.
- Polygenic disease: This is third category of genetic disorder. According to the polygenic hypothesis, many mild defects in genes conspire to produce some chronic diseases. To date, the full genetic basis of polygenic diseases has not been worked out as multiple interacting defects are highly complex.

In case of association analysis, we need to have selection of representative samples from the population of interest and complete and accurate genotype data set. Therefore, in this statistical analysis representative sample can be selected using appropriate sampling procedure depending on cost of experiments. However, missing values in genotypes is non-avoidable.

Therefore, we need to employ appropriate imputation techniques. Brief descriptions about these two techniques are given in subsequent paragraphs.

Sampling Techniques

The genesis of multiphase design for case control studies are from sample surveys. Initially, two phase sampling was introduced by Neyman (1938) as a technique for stratification. In this technique researcher needs to draw a Simple Random Sample from the target population and classify objects into homogeneous strata. Further, subsamples from each stratum are drawn and observations on variable are recorded only on these sub-samples drawn in the second phase. With judicious choice of strata and optimum sampling ratios, these designs are very cost efficient. The basic idea of these designs is to use information available on all subjects in the main study and draw more informative sub-samples for additional, more expensive, measurement and combining the information from both phases in the analysis. This concept for in Genome Wide Association Studies (GWAS) was introduced by Satagopan et. al. (2002),. Previously, this design has been cited in epidemiologic literature by White (1982). The basic goal of two phase design is to maximize the power to detect gene and disease association when the main design constraint is the total cost. Mainly, this total cost depends on number of gene evaluations rather than total number of individuals. Therefore, in the first phase, all the genes of our interest are evaluated on a sub-set of individuals. Later, most promising genes are evaluated on additional/same subjects in the second phase. This will eliminate the wastage of resources on genes, which are not likely to be associated with a particular trait. In this situation we find two types of cases i.e. (i) when genes are co-related (ii) when genes are independent.

Let us assume the unit cost per gene evaluation and let T denotes the total number of genetic evaluation or total cost. Let, in a genome, there are m genetic loci. Consider very simple situation that out of these m gene only one gene is associated with the trait (disease) under consideration. Now our problem is to identify this true gene which is associated with our trait. Let there are N individuals which are available. In absence of any cost constraints the best way is to evaluate all m markers for all N subjects with a total cost of mN. The best way of testing association with the trait under this situation is making 2 X 2 table for each locus with presence or absence of trait as rows and alleles as columns then apply chi-square test for association. The target gene would be selected based on largest test statistics. Now let us assume a situation when $T < mN$, then it is not possible to evaluate all m markers but only T/m individuals can be evaluated at first stage, but selection of T/m individuals should be in such that the possibility of missing true gene associated with trait is minimum. Therefore, this design needs to be optimized for two stage selection.

In this design, all m genes are evaluated on n_1 individuals, where proportion of cases with trait and control remains the same as in the case of N individuals. After application of test statistics, rank them based on absolute value of test statistics. In the second stage, select top mi genes

where "i" is proportion of genes on sub-sequent subjects till cost is T through selection of same proportion of case and control subject as in the original population. Now, the problem boils down to determination of value n_1 and i (ith proportion) so that it leads to maximum probability of selecting true gene i.e. maximum power P of the statistics. Then $T = n_1m + n_2mi$ where, n_2 is numbers of subjects at the second stage. Now our aim is to maximize P with respect to ni and i subject to fixed T and m. Since, $T/ m= n_1 + n_{2i}$ = fixed, therefore, choosing n_1 and i determines n_2. In other word, optimization of power for two stage design can be seen equivalently, as determination of proportion of resources at the first stage i.e. $j= n_1m/T$ and determination of proportion "i" of the genes to be evaluated at the second stage. The proportion of total number of subject required for two stage design is given by j+(1-j)/i.

In other words, P can be written as P1 * P2 under the assumption that mutational profiles of all genes are mutually uncorrelated. Hence, P1 is the probability that true gene is among top i^{th} proportion in stage 1 and P2 is the probability that true gene has highest association among all null genes at stage-2. These probabilities can be calculated using statistical distributions (may be Gaussian approximation). In practice, the assumption of independent gene outcome may not be true within individual in case of testing multiple markers. These outcomes may be correlated due to various factors, such as genetic linkages and loss of heterozygosis, allele frequency, and marker density. The correlation due to recombination can be easily quantified and further these can be modelled through statistical distributions. Further these probabilities can be further evaluated using Mote Carlo simulation for different values of i, j and μ (mean). This design can be further extended for optimal design for more than one true gene.

Genotype Imputation

Identification and characterization of genetic variation of a species which affects its important traits are very important for increasing production and productivity in agriculture especially in the context of development of improved biotic and abiotic resistance breeds/varieties. The basic idea is that, data on a modest set of genetic variant measured in number of related subjects can provide useful information about other genetic variants in those subjects forms the theoretical under pinning of both genetic linkage mapping in pedigree and haplotype mapping in founder population. These studies typically used few markers to survey entire genome through identification of parts of chromosomes inherited from common ancestor. Earlier in genetic linkage and haplotype mapping, it was expected that long sketches of shared chromosome inherited from a relatively recent common ancestor. Sometimes, the focus of GWAS is on unrelated individuals and it expected to have small stretch on shared chromosome. Under these circumstances genotype imputation can use these short stretches of shared haplotype to estimate with great precision the effects of many variants that are not directly genotyped.

There are two broad categories of genotype imputation. First, imputing missing genotype from information on close relatives and second, genotype imputation from distant relatives. If it is

known that the haplotype individuals carried at every point on the genome and SNP alleles are also known within each unique population haplotype then it is possible to impute genotypes which a individual carries for any SNP locus. Genotype imputation is important due to following reasons:-

- In case of accurate SNP array technology also, large number of SNP genotype are missing which poses problems in genomic selection and GWAS.
- Genotype imputation can be used to get high density genotype when subject has been genotyped with low density array.
- It is quite useful for combining data sets genotyped from two different panels with sufficient overlap between panels.
- Genotype imputation is applied to recover genotype from full genome sequence data. (i.e. from very dense SNP/insertion and deletion, CNV for genomic predictions and GWAS).

There are number of approaches/tools for imputing missing Genotypes such as PHASE (IMPUTE 1.0, IMPUTE 2.0) FastPHASE, MACH, BEAGLE etc. But PHASE and Fast PHASE are most widely used. In case of genotype imputation number of tools uses Hidden Marker Model (HMM) at the backend. These basic approaches relies that if it is known that a particular SNP alleles are associated with a particular hyplotype in a population then it is possible to infer or impute genotype carried by the individual of same hyplotype for which it is not known. In case of HMM, the hidden state generates true hyplotypes in the population for which genotypes are known. Then HMM can be used to estimate the probability that an individual carries a particular genotype at a particular locus given the genotype data for that individual at other locus and rest of the population. Basically it takes advantage of a reference population which is densely genotyped at all SNP. The methods of imputation differ in their assumptions about the hidden states, the way state transition probabilities are derived, emission probability and the initial state probabilities.

Association Analysis

The association analysis can be taken up with well-defined phenotype of a population, and genotypes data set which is collected using sound techniques. The preliminary analysis of genome-wide association data is a series of single-locus statistic tests, examining each SNP independently for association to the phenotype. The statistical test conducted depends on a variety of factors, but first and foremost, statistical tests are different for quantitative traits versus case/control studies. Quantitative traits are generally analysed using generalized linear model (GLM) approaches and most commonly the Analysis of Variance (ANOVA), which is similar to linear regression with a categorical predictor variable of genotype classes. The null hypothesis of an ANOVA using a single SNP is that there is no difference between the trait

means of any genotype group. The assumptions of GLM and ANOVA are that (i) the trait is normally distributed, (ii) the trait variance within each group is the same, and (iii) the groups are independent. Dichotomous case/control traits are generally analysed using either contingency table methods or logistic regression. Contingency table tests examine and measure the deviation from independence that is expected under the null hypothesis that there is no association between the phenotype and genotype classes using Chi-square test and related Fisher's exact test. Logistic regression is an extension of linear regression where the outcome of a linear model is transformed using a logistic function that predicts the probability of having case status given a genotype class. Logistic regression is often the preferred approach because it allows for adjustment for clinical covariates (and other factors), and can provide adjusted odds ratios as a measure of effect size. Logistic regression has been extensively developed, and numerous diagnostic procedures are available to aid interpretation of the model. For both quantitative and dichotomous trait analysis (regardless of the analysis method), there are a variety of ways that genotype data can be encoded or shaped for association tests. The choice of data encoding can have implications for the statistical power of a test, as the degrees of freedom for the test may change depending on the number of genotype-based groups that are formed. Allelic association tests examine the association between one allele of the SNP and the phenotype. Genotypic association tests examine the association between genotypes (or genotype classes) and the phenotype. The genotypes for a SNP can also be grouped into genotype classes or models, such as dominant, recessive, multiplicative, or additive models. Each model makes different assumptions about the genetic effect in the data by assuming two alleles for a SNP, A and a, a dominant model (for A) assumes that having one or more copies of the A allele increases risk compared to a (i.e. Aa or AA genotypes have higher risk). The recessive model (for A) assumes that two copies of the A allele are required to alter risk, so individuals with the AA genotype are compared to individuals with Aa and aa genotypes. The multiplicative model (for A) assumes that if there is 3 x risk for having a single A allele, there is a 9 x risk for having two copies of the A allele. In this case, if the risk for Aa is k, the risk for AA is k^2. The additive model (for A) assumes that there is a uniform, linear increase in risk for each copy of the A allele, so if the risk is 3 x for Aa, there is a 6x risk for AA. In this case, the risk for Aa is k and the risk for AA is 2k. A common practice for GWAS is to examine additive models only, as the additive model has reasonable power to detect both additive and dominant effects, but it is important to note that an additive model may be underpowered to detect some recessive effects. Rather than choosing one model a priori, some studies evaluate multiple genetic models coupled with an appropriate correction for multiple testing.

Acknowledgements

All contents of this lecture notes are taken from different web resources including research articles, presentations, lecture notes etc. We duly acknowledge contributions of all these resources.

Principal Component Analysis, Discriminant Analysis and Other Multivariate Statistical Techniques

Multivariate data consist of observations on several different variables for a number of individuals or subjects. Data of this type arise in all the branches of science, ranging from psychology to biology, and methods of analyzing multivariate data constitute an increasingly important area of statistics. Indeed, the vast majority of data in forestry is multivariate and proper handling of such data is highly essential. Principal components analysis (PCA) and Factor analysis (FA) are multivariate techniques applied to a single set of variables to discover which sets of variables in the set form coherent subsets that are relatively independent of one another. The details of PCA and FA are discussed as below.

Principal Components Analysis

Most of the times the variables under study are highly correlated and as such they are effectively "saying the same thing". To examine the relationships among a set of *p* correlated variables, it may be useful to transform the original set of variables to a new set of uncorrelated variables called *principal components*. These new variables are linear combinations of original variables and are derived in decreasing order of importance so that, for example, the first principal component accounts for as much as possible of the variation in the original data.

Let $x_1, x_2, x_3, \ldots, x_p$ are variables under study, then first principal component may be defined as

$$z_1 = a_{11} x_1 + a_{12} x_2 + ... + a_{1p} x_p$$

such that variance of z_1 is as large as possible subject to the condition that

$$a_{11}^2 + a_{12}^2 + ... + a_{1p}^2 = 1$$

This constraint is introduced because if this is not done, then Var(z_1) can be increased simply by multiplying any a_{1j}s by a constant factor

The second principal component is defined as

$$z_2 = a_{21} x_1 + a_{22} x_2 + ... + a_{2p} x_p$$

such that Var(z_2) is as large as possible next to Var(z_1)subject to the constraint that

$$a_{21}^2 + a_{22}^2 + ... + a_{2p}^2 = 1 \quad \text{and} \quad \text{cov}(z_1, z_2) = 0 \text{ and so on.}$$

It is quite likely that first few principal components account for most of the variability in the original data. If so, these few principal components can then replace the initial p variables in subsequent analysis, thus, reducing the effective dimensionality of the problem. An analysis of principal components often reveals relationships that were not previously suspected and thereby allows interpretation that would not ordinarily result. However, Principal Component Analysis is more of a means to an end rather than an end in itself because this frequently serves as intermediate steps in much larger investigations by reducing the dimensionality of the problem and providing easier interpretation. It is a mathematical technique which does not require user to specify the statistical model or assumption about distribution of original variates. It may also be mentioned that principal components are artificial variables and often it is not possible to assign physical meaning to them. Further, since Principal Component Analysis transforms original set of variables to new set of uncorrelated variables, it is worth stressing that if original variables are uncorrelated, then there is no point in carrying out principal component analysis.

Computation of principal components:

Let us consider the following data on average minimum temperature (x_1), average relative humidity at 8 hrs. (x_2), average relative humidity at 14 hrs. (x_3) and total rainfall in cm. (x_4) pertaining to Raipur district from 1970 to 1986 for kharif season from 21st May to 7th Oct.

X_1	x_2	x_3	x_4
25.0	86	66	186.49
24.9	84	66	124.34
25.4	77	55	98.79
24.4	82	62	118.88
22.9	79	53	71.88
7.7	86	60	111.96
25.1	82	58	99.74
24.9	83	63	115.20
24.9	82	63	100.16
24.9	78	56	62.38
24.3	85	67	154.40
24.6	79	61	112.71
24.3	81	58	79.63
24.6	81	61	125.59
24.1	85	64	99.87
24.5	84	63	143.56
24.0	81	61	114.97
Mean 23.56	82.06	61.00	112.97
S.D. 4.13	2.75	3.97	30.06

with the variance co-variance matrix.

$$\Sigma = \begin{bmatrix} 17.02 & -4.12 & 1.54 & 5.14 \\ & 7.56 & 8.50 & 54.82 \\ & & 15.75 & 92.95 \\ & & & 903.87 \end{bmatrix}$$

Find the Eigen values and eigen vectors of the above matrix. Arrange the eigen values in decreasing order. Let the eigen values in decreasing order and corresponding eigen vectors are

$\lambda_1 = 916.902 \quad a_1 = (0.006, \quad 0.061, \quad 0.103, \quad 0.993)$

$\lambda_2 = 18.375 \quad a_2 = (0.955, \quad -0.296, \quad 0.011, \quad 0.012)$

$\lambda_3 = 7.87 \quad a_3 = (0.141, \quad 0.485, \quad 0.855, \quad -0.119)$

$\lambda_4 = 1.056 \quad a_4 = (0.260, \quad 0.820, \quad -0.509, \quad 0.001)$

The principal components for this data will be

$$z_1 = 0.006\ x_1 + 0.061\ x_2 + 0.103\ x_3 + 0.993\ x_4$$

$$z_2 = 0.955\ x_1 - 0.296\ x_2 + 0.011\ x_3 + 0.012\ x_4$$

$$z_3 = 0.141\ x_1 + 0.485\ x_2 + 0.855\ x_3 - 0.119\ x_4$$

$$z_4 = 0.26\ x_1 + 0.82\ x_2 - 0.509\ x_3 + 0.001\ x_4$$

The variance of principal components will be eigen values i.e.

$Var(z_1) = 916.902, \ Var(z_2) = 18.375, \ Var(z_3) = 7.87, \ Var(z_4) = 1.056$

The total variation explained by original variables is

$$= Var(x_1) + Var(x_2) + Var(x_3) + Var(x_4)$$

$$= 17.02 + 7.56 + 15.75 + 903.87 = 944.20$$

The total variation explained by principal components is

$$\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 916.902 + 18.375 + 7.87 + 1.056 = 944.20$$

As such, it can be seen that the total variation explained by principal components is same as that explained by original variables. It could also be proved mathematically as well as empirically that the principal components are uncorrelated.

The proportion of total variation accounted for by the first principal component is

$$\frac{\lambda_1}{\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4} = \frac{916.902}{944.203} = .97$$

Continuing, the first two components account for a proportion

$$\frac{\lambda_1 + \lambda_2}{\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4} = \frac{935.277}{944.203} = .99$$

of the total variance.

Hence, in further analysis, the first or first two principal components z_1 and z_2 could replace four variables by sacrificing negligible information about the total variation in the system. The scores of principal components can be obtained by substituting the values of x_i s in equations of z_i s. For above data, the first two principal components for first observation i.e. for year 1970 can be worked out as

$z_1 = 0.006 \times 25.0 + 0.061 \times 86 + 0.103 \times 66 + 0.993 \times 186.49 = 197.380$

$z_2 = 0.955 \times 25.0 - 0.296 \times 86 + 0.011 \times 66 + 0.012 \times 186.49 = 1.383$

Similarly for the year 1971

$z_1 = 0.006 \times 24.9 + 0.061 \times 84 + 0.103 \times 66 + 0.993 \times 124.34 = 135.54$

$z_2 = 0.955 \times 24.9 - 0.296 \times 84 + 0.011 \times 66 + 0.012 \times 124.34 = 1.134$

Thus the whole data with four variables can be converted to a new data set with two principal components.

Note: The principal components depend on the scale of measurement, for example, if in the above example X_1 is measured in ^0F instead of ^0C and X_4 in mm in place of cm, the data gives different principal components when transformed to original x's. In very specific situations results are same. The conventional way of getting around this problem is to use standardized variables with unit variances, i.e., correlation matrix in place of dispersion matrix. But the principal components obtained from original variables as such and from correlation matrix will not be same and they may not explain the same proportion of variance in the system. Further more, one set of principal components is not simple function of the other. When the variables are standardized, the resulting variables contribute almost equally to the principal components determined from correlation matrix. Variables should probably be standardized if they are measured on scales with widely differing ranges or if measured units are not commensurate. Often population dispersion matrix or correlation matrix are not available. In such situations sample dispersion matrix or correlation matrix can be used.

Applications of principal components:

- The most important use of principal component analysis is reduction of data. It provides the effective dimensionality of the data. If first few components account for most of the variation in the original data, then first few components' scores can be utilized in subsequent analysis in place of original variables.
- Plotting of data becomes difficult with more than three variables. Through principal component analysis, it is often possible to account for most of the variability in the data by first two components, and it is possible to plot the values of first two components scores for each individual. Thus, principal component analysis enables us to plot the data in two dimensions. Particularly detection of outliers or clustering of individuals will be easier through this technique. Often, use of principal component analysis reveals grouping of variables which would not be found by other means.
- Reduction in dimensionality can also help in analysis where no. of variables is more than the number of observations, for example, in discriminant analysis and regression analysis. In such cases, principal component analysis is helpful by reducing the dimensionality of data.
- Multiple regression can be dangerous if independent variables are highly correlated. Principal component analysis is the most practical technique to solve the problem. Regression analysis can be carried out using principal components as regressors in place of original variables. This is known as principal component regression.

Discriminant Analysis

Discriminant analysis and classification are multivariate techniques concerned with separating distinct sets of objects (or observations) and with allocating new objects (observations) to previously defined groups. Discriminant analysis is rather exploratory in nature. As a separatory procedure, it is often employed on a one - time basis in order to investigate observed differences when causal relationships are not well understood. Classification procedures are less explanatory in the sense that they lead to well- defined rules, which can be used for assigning new objects. Classification ordinarily requires more problem structure than discrimination.

Thus, the immediate goals of discrimination and classification, respectively, are as follows.

Goal 1. To describe either graphically (in three or lower dimensions) or algebraically, the differential features of objects (observations) from several known collections (populations). We try to find "discriminants" whose numerical values are such that the collections are separated as much as possible.

Goal 2. To sort objects (observations) into two or more labeled classes. The emphasis is on deriving a rule that can be used to optimally assign a new object to the labeled classes.

We shall follow convention and use the term discrimination to refer to Goal 1. This terminology was introduced by R.A. Fisher in the first modern treatment of separatory problems. A more descriptive term for this goal, however, is separation, we shall refer to the second goal as classification, or allocation.

A function that separates may sometimes serve as an allocation, and conversely, an allocatory rule may suggest a discriminatory procedure. In practice, Goals 1 and 2 frequently overlap and the distinction between separation and allocation becomes blurred.

Here we discuss Fisher's linear discriminant function for two multivariate populations having same dispersion matrix. For more general cases readers are requested to go through the references cited at the end.

Fisher's Discriminant Function

Here Fisher's idea was to transform the multivariate observations **x** to univariate observations y such that the y's derived from populations π_1 and π_2 were separated as much as possible. Fisher's approach assumes that the populations are normal and also assumes the population covariance matrices are equal because a pooled estimate of common covariance matrix is used.

A fixed linear combination of the **x**'s takes the values $y_{11}, y_{12}, ..., y_{1n1}$, for the observations from the first population and the values $y_{21}, y_{22}, ..., y_{2n2}$, for the observations from the second population. The separation of these two sets of univariate y's is assessed in terms of the differences between $\bar{y}_1$ and $\bar{y}_2$ expressed in standard deviation units. That is,

$$\text{separation} = \frac{|\bar{y}_1 - \bar{y}_2|}{s_y}, \text{ where } s_y^2 = \frac{\sum_{j=1}^{n1}(y_{1j} - \bar{y}_1)^2 + \sum_{j=1}^{n2}(y_{2j} - \bar{y}_2)^2}{n_1 + n_2 - 2}$$

is the pooled estimate of the variance. The objective is to select the linear combination of the **x** to achieve maximum separation of the sample means $\bar{y}_1$ and $\bar{y}_2$.

Result: The linear combination $y = \hat{\boldsymbol{l}}'\mathbf{x} = (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)'\mathbf{S}_{pooled}^{-1}\mathbf{x}$ maximizes the ratio

$$\frac{(\text{Squared distance between sample means of y})}{(\text{Sample variance of y})} = \frac{(\bar{y}_1 - \bar{y}_2)^2}{s_y^2}$$

$$= \frac{(\hat{l}'\bar{x}_1 - \hat{l}'\bar{x}_2)^2}{\hat{l}'S_{pooled}\hat{l}} = \frac{(\hat{l}'d)^2}{\hat{l}'S_{pooled}\hat{l}}$$

overall possible coefficient vectors $\hat{l}'$ where $d = (\bar{x}_1 - \bar{x}_2)$. The maximum of the above ratio is $D^2 = (\bar{x}_1 - \bar{x}_2)' s_{pooled}^{-1} (\bar{x}_1 - \bar{x}_2)$, the Mahalanobis distance.

Fisher's solution to the separation problem can also be used to classify new observations. An allocation rule is as follows.

Allocate x_0 to π_1 if

$$y_0 = (\bar{x}_1 - \bar{x}_2)' s_{pooled}^{-1} x_0 \geq \hat{m} = \frac{1}{2}(\bar{x}_1 - \bar{x}_2)' s_{pooled}^{-1} (\bar{x}_1 + \bar{x}_2)$$

and to π_2 if

$$y_0 < \hat{m}$$

If we assume the populations π_1 and π_2 are multivariate normal with a common covariance matrix, then a test of $H_0: \mu_1 = \mu_2$ versus $H_1: \mu_1 \neq \mu_2$ are accomplished by referring

$$\frac{(n_1 + n_2 - p - 1)}{(n_1 + n_2 - 2)p} \left(\frac{n_1 n_2}{n_1 + n_2} \right) D^2$$

to an F-distribution with $\upsilon_1 = p$ and $\upsilon_2 = n_1 + n_2 - p - 1$ d.f. If H_0 is rejected, we can conclude the separation between the two populations is significant.

Example:

To construct a procedure for detecting potential hemophilia 'A' carriers, blood samples were analyzed for two groups of women and measurements on the two variables, $x_1 = \log_{10}$(AHF activity) and $x_2 = \text{loh}_{10}$(AHF-like antigens) recorded. The first group of $n_1 = 30$ women were selected from a population who do not carry hemophilia gene (normal group). The second group of $n_2 = 22$ women were selected from known hemophilia 'A' carriers (obligatory group). The mean vectors and sample covariance matrix are given as

$$\bar{x}_1 = \begin{bmatrix} -0.0065 \\ -0.0390 \end{bmatrix}, \quad \bar{x}_2 = \begin{bmatrix} -0.2483 \\ 0.0262 \end{bmatrix} \text{ and } S_{pooled}^{-1} = \begin{bmatrix} 131158 & -90.423 \\ -90.423 & 108.147 \end{bmatrix}$$

Now the linear discriminant function is

$$y_0 = \hat{l}' x_0 = (\bar{x}_1 - \bar{x}_2)' S_{pooled}^{-1} x_0$$

$$= \begin{bmatrix} .2418 & -0.0652 \end{bmatrix} \begin{bmatrix} 131158 & -90.423 \\ -90.423 & 108.147 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$= 37.61x_1 - 28.92\, x_2$$

Moreover

$$\bar{y}_1 = \hat{l}'\, \bar{\mathbf{x}}_1 = \begin{bmatrix} 37.61 & -28.92 \end{bmatrix} \begin{bmatrix} -0.0065 \\ -0.0390 \end{bmatrix} = 0.88$$

$$\bar{y}_2 = \hat{l}'\, \bar{\mathbf{x}}_2 = \begin{bmatrix} 37.61 & -28.92 \end{bmatrix} \begin{bmatrix} -0.2483 \\ -0.0262 \end{bmatrix} = -10.10$$

and the mid-point between these means is

$$\hat{\mathbf{m}} = \frac{1}{2}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)' \mathbf{s}_{\text{pooled}}^{-1} (\bar{\mathbf{x}}_1 + \bar{\mathbf{x}}_2) = \frac{1}{2}(\bar{y}_1 + \bar{y}_2) = -4.61$$

Now to classify a women who may be a hemophilia 'A' carrier with $x_1 = -.210$ and $x_2 = -0.044$, we calculate

$y_0 = \hat{l}'\, \mathbf{x}_0 = 37.61x_1 - 28.92\, x_2 = -6.62$

Since $y_0 < \hat{m}$ we classify the women in π_2 population, i.e., to obligatory carrier group.

Correspondence Analysis

The past decade has seen tremendous growth in the availability of both computer hardware and statistical software. As a result, the use of multivariate statistical techniques has increased to include most fields of scientific research and many areas of business and public management. In both research and management domains there is increasing recognition of the need to analyze data in a manner that takes into account the interrelationships among variables. Variables can be classified as being quantitative or qualitative. A quantitative variable is the one in which the variates differ in magnitude, for example, income, age and weight. A qualitative variable is one in which the variates differ in kind rather than in magnitude, for example, marital status, sex, nationality and hair colour. Obtaining values for quantitative variables involves measurement along a scale and unit of measure. A unit of measure may be infinitely divisible (for example, kilometers, meters, etc.) or indivisible (for example, family size.). When the units of measure are infinitely divisible the variable is said to be continuous. In the case of an indivisible of unit of measure the variable is said to be discrete.

Scales of measurement can also be classified on the basis of the relations among the elements composing the scale. For example, an ordinal scale is the one in which the elements along the scale can be ordered from low to high. A nominal scale corresponds to qualitative data. An example would be the variable *marital status* which has the categories married, single, divorced, widowed and separated. The five categories can be assigned coded values such as 1, 2, 3, 4, or 5. Although these coded values are numerical, they must not be treated as quantitative. On occasion, quantitative variables are treated in an analysis as if they were nominal. In general, we use the term *categorical* to denote a variable that is used as if it was nominal. The variable age for example can be divided into 6 levels and coded 1, 2, 3, 4, 5, and 6. Principal component analysis and Factor analysis are primarily designed for analysis of data on continuous variables, whereas correspondence analysis is designed for categorical data. Before going in detail for correspondence analysis, we explain few terms that are commonly used in it.

Two-Dimensional Contingency Tables: In the event, a sample of n observations is simultaneously cross-classified with respect to the two categorical random variables (X, Y) the joint frequencies can be summarized in a table called two-dimensional contingency table.

	Y						
X		1	2	3	…	c	Total
	1	f_{11}	f_{12}	f_{13}	…	f_{1c}	$f_{1.}$
	2	f_{21}	f_{22}	f_{23}	…	f_{2c}	$f_{2.}$
	3	f_{31}	f_{32}	f_{33}	…	f_{3c}	$f_{3.}$
	. . .						
	R	f_{r1}	f_{r2}	f_{r3}	…	f_{rc}	$f_{r.}$
	Total	$f_{.1}$	$f_{.2}$	$f_{.3}$	…	$f_{.c}$	1.00

The random variable X is assumed to have a range of values consisting of *r* categories, whereas the variable Y is assumed to have *c* categories. The cell density or joint density for cell (i, j) is denoted by f_{ij}, i = 1, 2, …, r; j= 1, 2, …, c; where it is understood that the first subscript refers to the row and the second subscript to the column. The marginal densities are denoted by $f_{i.}$ and $f_{.j}$ for the row and column variables respectively. The conditional densities for the rows given column *j* will be denoted by $f_{i.}(i|j)$ and for the columns given row *i* by $f_{.j}(j|i)$.

Row and column proportions

The conditional densities $f_{.j}(j|i)$ are often referred to as *row proportions*, and the marginal density $f_{.j}$ is called the *column total proportions*. In a similar fashion the conditional densities $f_{i.}(i|j)$ are often referred to as *column proportions*, and the marginal density $f_{i.}$is called the *row total proportions*.

Row and column profiles:

The row and column proportions are also commonly referred to as *row* and *column profiles.* The term profile is often used in connection with the graphical displays of relationships in a contingency table.

Singular value decomposition (*SVD*): A real (n × p) matrix **A** of rank *k* can be expressed as the product of three matrices that have a useful interpretation. This decomposition of **A** is referred to as a *singular value decomposition* and is given by

A = **UDV'**

where

1. **D** (k × k) is a diagonal matrix with positive diagonal elements $\alpha_1, \alpha_2, \cdots, \alpha_k$, which are called the singular values of **A**, (without loss of generality we assume that the α_j, j = 1, 2, …, k, are arranged in descending order).

2. The k columns of **U** (n × k), $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k$, are called the *left singular vectors* of **A** and the k columns of **V** (p × k), $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k$, are called the *right singular vectors* of **A**

3. The matrix **A** can be written as the sum of k matrices, each with rank 1, $\mathbf{A} = \sum_{j=1}^{k} \boldsymbol{\alpha}_j \mathbf{u}_j \mathbf{v}_j'$. The subtraction of any one of these terms from the sum results in a singular matrix for the remainder of the sum.

4. The matrices **U** (n × k), and **V** (p × k) have the property that **U'U** = **VV' = I**; hence the columns of **U** form an orthonormal basis for the columns of **A** in n-dimensional space and the columns of **V** form an orthonormal basis for the rows of **A** in p-dimensional space.

5. Let **A**(*l*) denote the first *l* terms of the singular value decomposition for **A**; hence **A**(*l*) = $\sum_{j=1}^{l} \boldsymbol{\alpha}_j \mathbf{u}_j \mathbf{v}_j'$. This expression minimizes *tr*[(**A** – **X**) (**A** – **X**)'] = $\sum_{i=1}^{n}\sum_{j=1}^{p}(a_{ij} - x_{ij})^2$ among all (n × p) matrices **X** of rank *l*. Thus the singular value decomposition can be used to provide a *matrix approximation* to **A**.

Biplots: A biplot is used to provide a two-dimensional representation for a data matrix **X**. Only two dimensions are usually employed to keep the presentation simple. It is assumed that a singular value decomposition approximation for **X** based on *r* = 2 dimensions is adequate. This of course should be evaluated by examining the magnitudes of the singular

values beyond $r = 2$. The sum of these remaining residual singular values should ideally represent only a small proportion of $tr\mathbf{D}$.

A singular value decomposition approximation for $\mathbf{X}$ based on two-dimensions is given by $\hat{\mathbf{X}} = \mathbf{U}_1\mathbf{D}_1\mathbf{V}_1'$, where the rows of $\mathbf{V}_1'(2 \times p)$ are the eigen vectors of $\mathbf{X'X}$ and the columns of $\mathbf{U}_1(n \times 2)$ are the eigen vectors of $\mathbf{XX'}$. There are several ways of employing the three elements of the right hand side of the equation for $\hat{\mathbf{X}}$. The most common form which is called the *principal components plot*.

Correspondence analysis is a technique that uses singular value decomposition to analyze a matrix of nonnegative data. The technique simultaneously characterizes the relationship among the rows and also among the columns of the data matrix. The outcome of a correspondence analysis is a pair of *bivariate plots*. One bivariate plot is based on the first two principal axes derived from the row profiles, and the second plot is based on the first two principal axes obtained from the column profiles. Points representing the row categories are plotted using the row principal axes and points representing the column categories are plotted using the column principal axes. The spatial relationships among the two sets of categories can then be studied using the two bivariate plots. By using the same pair of axes to denote both pairs of principal axes the two bivariate plots can be superimposed on one another. With both plots appearing on the same axes the spatial relationship between the row categories and column categories can also be related. The SAS computer software procedure CORRESP will be used throughout this section to perform the necessary data analysis.

Table 1: Correspondence matrix of observed cell densities for an ($\mathbf{r} \times \mathbf{c}$) contingency table

	1	2	3		C	Row Masses
1	O_{11}	O_{12}	O_{13}		O_{1c}	$O_{1.}$
2	O_{21}	O_{22}	O_{23}		O_{2c}	$O_{2.}$
3	O_{31}	O_{32}	O_{33}		O_{3c}	$O_{3.}$
r	O_{r1}	O_{r2}	O_{r3}		o_{rc}	$O_{r.}$
Column Masses	$O_{.1}$	$O_{.2}$	$O_{.3}$		$O_{.c}$	1

Correspondence analysis for two-dimensional contingency tables

Correspondence analysis can be used to study interaction in a two-dimensional contingency table. Table 1 shows the observed *cell proportions* or *cell densities*. Let us denote the cell density for cell (i,j) as $O_{ij}=n_{ij}/n$, where n_{ij} denotes the sample frequency in cell (i,j); $i = 1,2, \ldots, r$ and $j = 1,2, \ldots, c$. The row and column marginal densities are given by $O_{i.} = n_{i.}/n$ and

$O_{.j}=n_{.j}/n$ respectively where $n_{i.}$ and $n_{.j}$ are the row and column marginal frequencies respectively.

Example: The data given in Table 2 pertains to the student-run legal advice service for the poor. This table examines the relationship between the type of criminal charge and the eventual outcome of the case for both males and females.

Table 2. Contingency Table for Criminal Charge Data

Convicted	Sex	Charge: Impaired Driving	Theft Under $ 1000	Mischief	Possession of Narcotics	other	Totals
No	Male	8	11	5	7	12	43
	Female	5	15	3	1	6	30
Yes	Male	105	32	11	23	37	208
	Female	32	57	6	2	25	122
Totals		150	115	25	33	80	403

Table 3. Correspondence Matrix for Criminal Charge Data

Convicted	Sex	Impaired Driving	Theft Under $ 1000	Mischief	Possession of Narcotics	other	Totals
No	Male	2.0	2.7	1.2	1.7	3.0	10.7
	Female	1.2	3.7	0.7	0.2	1.5	7.4
Yes	Male	26.1	7.9	2.7	5.7	9.2	51.6
	Female	7.9	14.1	1.5	0.5	6.2	30.3
Column Mass		37.2	28.6	6.2	8.1	19.9	100.0

The corresponding matrix of cell densities and row and column marginal densities are shown in Table 3. The numbers are given as percentages and hence represent 100 $O_{ij.}$The column of row masses on the right presents the row marginals as percents 100 $O_{i.}$, and the row of column masses (last row) displays the column marginals, 100 $O_{.j.}$ The majority of the clients were convicted males and 30.3% of the samples were convicted females. The two most common offences were impaired driving (37.2%) and theft under $1000 (28.6%). The most

common offence for males was impaired driving (28.1% of the sample) and the most common female offence was theft under $1000 (17.8% of the sample).

Correspondence Matrix and Row and Column Masses

The (r×c) matrix of cell densities as shown in Table 1 is denotes by **O** and is called the *Correspondence Matrix.* The (r×1) vector of row marginals $O_{i.}$, i =1,2…,r, is denoted by r and similarly the (c×1) vector of column marginals $O_{.j}$, j=1,2…,c, is denoted by c. *These row and column marginal vectors* can be written as $\mathbf{r} = \mathbf{O}\mathbf{e}_c$ and $\mathbf{c} = \mathbf{O}'\mathbf{e}_r$ where $\mathbf{e}_c$(c×1) and $\mathbf{e}_r$ (r×1) are vectors of unities. The vectors **r** and **c** are also referred to respectively as *row and column Masses*. Diagonal matrices constructed from the row and

Table 4. Matrix R for Row Profiles

		Columns					
		1	**2**	**3**	**…**	**c**	**Totals**
Rows	1	$n11/n_{1.}$	$n_{12}/n_{1.}$	$n_{13}/n_{1.}$	…	$n_{1c}/n_{1.}$	1
	2	$n_{21}/n_{2.}$	$n_{22}/n_{2.}$	$n_{33}/n_{2.}$	…	$n_{2c}/n_{2.}$	1
	3	$n_{31}/n_{3.}$	$n_{32}/n_{3.}$	$n_{33}/n_{3.}$	…	n_{3c}/n_3	1
	⋮	⋮	⋮	⋮		⋮	⋮
	R	$n_{r1}/n_{r.}$	$n_{r2}/n_{r.}$	$n_{r3}/n_{r.}$	…	$n_{rc}/n_{r.}$	1
	Column Mass	$n_{.1}/n$	$n_{.2}/n$	$n_{.3}/n$	…	$n_{.c}/n$	1

column masses are denoted by $\mathbf{D}_r$ (r×r) and $\mathbf{D}_c$ (c×c) respectively. The diagonal elements of $\mathbf{D}_r$ are the elements of **r** and the diagonal elements of $\mathbf{D}_c$ are the elements of **c**.

Row and Column Profiles

Beginning with the table of cell frequencies n_{ij} for each row i, the (c×1) vector of row conditional densities is determined from $n_{ij}/n_{i.}$, j=1,2,…, c, and is denoted by $\mathbf{r}_i$. These row conditional densities are called *row profiles.* The complete set of r row profiles will be denoted by the (r×c) matrix **R** with rows given by $\mathbf{r}_{i.}$, i =1,2,…,r. Similarly the vector of column conditional densities $n_{ij}/n_{.j}$, i=1,2…,r, for column j is denoted by the (r×1) vector $\mathbf{c}_j$, j = 1,2, …,c. The matrices **R** and **C** are illustrated in Tables 4 and 5 respectively. These row and column profile matrices are useful to judge the departure from independence. For the criminal charge data the row profile matrix R and the column profile matrix C are summarized in Tables 6 and 7. The row profiles in Table 6.

Table 5. Matrix C for Column Profiles

		Columns					
		1	**2**	**3**	**…**	**C**	**Row Mass**
Rows	1	$n11/n._1$	$n_{12}/n._1$	$n_{13}/_{n.1}$	…	$n_{1c}/n._c$	$N_{1.}/n$
	2	$n_{21}/n._2$	$n_{22}/n._2$	$n_{33}/n._2$	…	$n_{2c}/n._c$	$N_{2.}/n$
	3	$n_{31}/n._3$	$n_{32}/n._3$	$n_{33}/n._3$	…	$n_{3c}/n._c$	$N_{3.}/n$
	⋮	⋮	⋮	⋮		⋮	⋮
	R	$n_{r1}/n._1$	$n_{r2}/n._2$	$n_{r3}/n._3$	…	$n_{rc}/n._c$	$n_{r.}/n$
	Totals	1	1	1	…	1	1

compare the four sex/conviction categories. The tow female profiles (no and yes) are quite similar to each other, but the two male profiles are different from each other. For the column profiles in Table 7 the impaired driving and possession of narcotics profiles are similar to each other. Also the mischief and other profiles are similar. The profile for the theft under $1000 is quite different from the other four column profiles. Since theft under $1000 is the only offence dominated by females we shall see that this provides a partial explanation for this different column profile.

Table 6. Row Profiles for Criminal Charge Data

		Charge					
Convicted	Sex	Impaired Driving	Theft Under $ 1000	Mischief	Possession of Narcotics	other	Totals
No	Male	0.186	0.256	0.116	0.163	0.279	1.000
	Female	0.167	0.500	0.100	0.033	0.200	1.000
Yes	Male	0.505	0.154	0.053	0.111	0.178	1.000
	Female	0.262	0.467	0.049	0.016	0.205	1.000
Column Mass		0.372	0.286	0.062	0.081	0.199	1.000

Table 7. Column Profiles for Criminal Charge Data

		Charge					
Convicted	Sex	Impaired Driving	Theft Under $ 1000	Mischief	Possession of Narcotics	other	Row Mass
No	Male	0.053	0.096	0.200	0.212	0.150	0.107
	Female	0.033	0.130	0.120	0.030	0.075	0.074
Yes	Male	0.700	0.278	0.440	0.697	0.463	0.516
	Female	0.214	0.496	0.240	0.061	0.312	0.303
		1.000	1.000	1.000	1.000	1.000	1.000

Departure from Independence

The purpose of correspondence analysis in the study of contingency tables is usually to study the departure of the observed cell frequencies from the cell frequencies expected under independence. Although it is possible to compare the observed cell frequencies from other models, the independence model is the most commonly used base for comparisons. Under the independence assumption, the theoretical row profiles for each row should be equal to the column marginals and equivalently the true column profiles for each column should be equal to the row marginals.

Table 8. Row profile deviation from independence

Charge						
Convicted	**Sex**	**Impaired driving**	**Theft under $1000**	**Mischief**	**Possession of Narcotics**	**Other**
No	Male	-0.186	-0.030	0.054	0.082	0.080
	Female	-0.205	0.314	0.038	-0.048	0.001
Yes	Male	0.133	-0.132	-0.009	0.030	-0.021
	Female	-0.110	0.181	-0.013	-0.065	0.006

Table 9. Column profile deviation from independence

Charge						
Convicted	**Sex**	**Impaired driving**	**Theft under $1000**	**Mischief**	**Possession of Narcotics**	**Other**
No	Male	-0.054	-0.011	0.093	0.105	0.043
	Female	-0.041	0.056	0.046	-0.044	0.001
Yes	Male	0.284	-0.238	-0.076	0.181	-0.053
	Female	-0.089	0.193	-0.063	-0.242	0.009

For the sample correspondence matrix therefore the matrix differences (**R-e**$_r$**c**') and (**C-re'**$_c$) measure the degree of departure or deviation from independence in the sample (Tables 8 and 9). Equivalently, under independence the cross product of the sample row and column marginal vectors or masses should be approximately equal to the correspondence matrix **O** of observed cell densities. The matrix difference (**O-rc**') is also therefore a measure of the deviation from independence.

Pearson Chi-Square Statistic and Total Inertia: The Pearson Chi-Square Statistic for testing independence is given as

$$G^2 = \sum_{i=1}^{r}\sum_{j=1}^{c} \frac{(n_{ij} - n_{i.}n_{.j}/n)^2}{n_{i.}n_{.j}/n}$$

or $$G^2 = \sum_{i=1}^{r} n_{i.}[\sum_{j=1}^{c} (\frac{n_{ij}}{n_{i.}} - n_{.j}/n)^2 /(n_{.j}/n)]$$

or $$G^2 = \sum_{j=1}^{c} n_{.j}[\sum_{i=1}^{r} (\frac{n_{ij}}{n_{.j}} - n_{i.}/n)^2 /(n_{i.}/n)]$$

The above versions of the Pearson Chi-Square Statistic can also be expressed as

$$G^2 = \sum_{i=1}^{r} n_{i.}(\mathbf{r}_i - \mathbf{c})'\mathbf{D}_\mathbf{c}^{-1}(\mathbf{r}_i - \mathbf{c})$$

or $$G^2 = \sum_{j=1}^{c} n_{.j}(\mathbf{C}_j - \mathbf{r})'\mathbf{D}_\mathbf{r}^{-1}(\mathbf{C}_j - \mathbf{r}).$$

The statistic G^2 / n is called the *total inertia*. Further, it can be viewed as a measure of the magnitude of the total row squared deviations or equivalently the magnitude of the column squared deviations. Total inertia can also be expressed in the form

$$Tr[\,\mathbf{D}_\mathbf{r}^{-1}(\mathbf{O} - \mathbf{r}\mathbf{c}')\mathbf{D}_\mathbf{c}^{-1}(\mathbf{O} - \mathbf{r}\mathbf{c}')'\,]$$

Table 10: Contribution to Chi-Square statistic for criminal charge data

Charge							
Convicted	**Sex**	**Impaired driving**	**Theft under $1000**	**Mischief**	**Possession of Narcotics**	**Other**	**Totals**
No	Male	4.00	0.13	2.04	3.44	1.41	11.02
	Female	3.41	4.84	0.70	0.86	0.00	9.81
Yes	Male	9.83	12.61	0.28	2.09	0.44	25.25
	Female	3.96	14.14	0.32	6.39	0.03	24.84
Totals		21.20	31.72	3.34	12.78	1.88	70.92

Table 10 shows the cell contributions to the total Chi-square statistic.

Coordinates of row and column profiles: For the singular value decomposition of $(\mathbf{O} - \mathbf{rc}')$ given by $\mathbf{AD}_\mu\mathbf{B}'$ the columns of matrices **A** and **B** provide the principal axes for the columns and rows of $(\mathbf{O} - \mathbf{rc}')$ respectively. Each row of $(\mathbf{O} - \mathbf{rc}')$ can be expressed as a linear combination of the rows of $\mathbf{B}'$ (columns of **B**), and hence the coordinates for the rows of $(\mathbf{O} - \mathbf{rc}')$ in the space generated by the rows of $\mathbf{B}'$ are given by the $\mathbf{AD}_\mu$. The coordinates for the *i*th row of $(\mathbf{O} - \mathbf{rc}')$ are given by the *i*th row of $\mathbf{AD}_\mu$. Similarly the coordinates for the columns of $(\mathbf{O} - \mathbf{rc}')$ with respect to the space generated by the columns of **A** are provide by the columns of $\mathbf{D}_\mu\mathbf{B}'$.

To obtain the coordinates for the row and column profile deviations, the relationships

$$(\mathbf{R} - \mathbf{e}_r\mathbf{c}') = \mathbf{D}_r^{-1}(\mathbf{O} - \mathbf{rc}')$$

and

$$(\mathbf{C} - \mathbf{re}_c') = \mathbf{D}_c^{-1}(\mathbf{O} - \mathbf{rc}')$$

can be used. The required coordinates for the row and column profile deviations are therefore given by

$$\mathbf{V}\ (r \times k) = \mathbf{D}_r^{-1}\mathbf{AD}_\mu = \mathbf{D}_r^{-1}(\mathbf{O} - \mathbf{rc}')\ \mathbf{D}_c^{-1}\mathbf{B}$$

and

$$\mathbf{W}\ (c \times k) = \mathbf{D}_c^{-1}\mathbf{BD}_\mu = \mathbf{D}_c^{-1}(\mathbf{O} - \mathbf{rc}')'\ \mathbf{D}_r^{-1}\mathbf{A}$$

respectively.

The coordinates for row profiles on row principal axes and coordinates for column profiles on column principal axes are given in Tables 11 and 12.

Table 11: Coordinates for row profiles on row principal axes

		Row principal axes (columns of B)				
		1	**2**	**3**	**…**	**K**
Rows	1	v_{11}	v_{12}	v_{13}	…	v_{1k}
	2	v_{21}	v_{22}	v_{23}	…	v_{2k}
	3	v_{31}	v_{32}	v_{33}	…	v_{3k}
	⋮	⋮	⋮	⋮		⋮
	r	v_{r1}	v_{r2}	v_{r3}	…	v_{rk}

Table 12. Coordinates for column profiles on column principal axes

		Column principal axes (columns of A)				
		1	2	3	…	K
Rows	1	w_{11}	w_{12}	w_{13}	…	w_{1k}
	2	w_{21}	w_{22}	w_{23}	…	w_{2k}
	3	w_{31}	w_{32}	w_{33}	…	w_{3k}
	⋮	⋮	⋮	⋮		⋮
	c	w_{c1}	w_{c2}	w_{c3}	…	w_{ck}

For the criminal charge data the coordinates for the row and column profile deviations on their respective dimensions are shown in Tables 13 and 14. For the row profiles it would appear that the first dimension reflects a contrast between females charged and males convicted. The second row dimension is primarily a measure of males charged but not.

Table 13. Coordinates for row profiles on row principal axes for criminal charge data

Row Profile		**Principal axes**		
		1	**2**	**3**
No	Males	0.04	0.50	-0.03
No	Females	0.55	0.09	0.11
Yes	Males	-0.35	-0.05	0.01
Yes	Females	0.44	-0.11	-0.03

Table 14. Coordinates for column profiles on column principal axes for criminal charge data

Column Profile	**Principal axes**		
	1	**2**	**3**
Impaired driving	-0.34	-0.16	0.01
Theft under $1000	0.52	-0.06	0.00
Mischief	0.08	0.34	0.11
Possession of Narcotics	-0.50	0.37	-0.01
Other	0.07	0.13	-0.05

convicted. For the column profiles the first dimension represents a contrast between the theft under $1000 and the crimes of narcotics possession and impaired driving. The second dimension for the column profile deviations seems to reflect a contrast between the three charges mischief, narcotics possession and other offences with the charge impaired.

Factor Analysis

Factor analysis is a data reduction technique, which often requires large sample size to have a valid interpretation. The basic idea in factor analysis is that a large number of explanatory variables having similar type of responses can be captured with a single latent variable that cannot be measured directly. For example, the latent variable (or factor) socioeconomic status is associated with the observed variables income, education, health status, occupation, on which the peoples' responses are of similar type.

In factor analysis, the number of factors is same as the number of variables, where each factor captures a certain amount of variation of all the variations present in the observed variables. The factors are always arranged in the decreasing order of their variances. In factor analysis, one expects three outputs viz., common factor variances, factor loadings and factor scores. The common factor variance is the measure of the amount variation explained by a factor present in the observed variables. Factor loading measures the underlying relationship that an observed variable have with a factor. The factor scores are the transformed data, commonly the weighted sum/mean of the observed variables (or manifest variables).

The factor scores are not the penultimate output rather than act as an intermediate step (dimensionality reduction) for carrying out further statistical analysis, a much important one. In other words, factor scores enable user to use a single variable, instead of set of variables, as a measure of the factor in the other statistical investigation. For example, in case of linear model or mixed model, the factor scores can be used as variable (fixed factors or random factors), but here it refers to the categorical independent variable. Further, technically the factor scores are continuous and hence can be used as covariates in the model rather than as factors.

Type of Factor Analysis

There are two types of factor analysis, one is Exploratory Factor Analysis (EFA) and other is Confirmatory Factor Analysis (CFA). In CFA, one assumption is that there should be prior information about the number of factors likely to be encountered as well as which variables will be loaded onto which factors. On the other hand, CFA allows the researchers to test the hypothesis that whether the relationship between a variable and the underlying factor exits or not. Initially, the researcher postulates a certain a priori relationship pattern based on existing knowledge i.e., published research (empirical and/or theoretical) and then test the hypothesis

statistically. In EFA, the researcher tries to find out the number of underlying constructs (factors) without having any a priori information about the number of factors. In other words, in EFA, the number of factors is determined on the basis of the dataset supplied by the user, and also depends upon user interpretation. Linking these two approaches, one can use EFA first to explore the underlying factors and then perform CFA to validate the structure of factors in a new dataset that has not been used for performing EFA. For example, a factor "depression" can be obtained with underlying variables depressed mood, fatigue, exhaustion and social dysfunction through EFA for a sample of rural women, and then the CFA can be used to validate this factor using a sample of urban women. In EFA, the cut-off of loading are much relaxed than that of CFA. In other words, a variable having loading value <|0.7| is disqualified from its loading onto a certain factor (Thumb rule). Generally, the EFA is most commonly used in day-to-day life than that of CFA. So, in this study material we only focused on EFA.

Exploratory Factor Analysis (EFA)

Before carrying out factor analysis, some important points need to be considered. At first, the reliability of the dataset should be checked for factor analysis. In other words, for factor analysis, the values of the variables should be in interval scale, each variable should be normally distributed, pairs of variables should follow bi-variate normal distribution and the dataset as a whole should follow multivariate normal distribution. Further, the sample size should be large. Field (2000) suggested 10-15 observations per variable. Habing (2003) state that there should be at least 50 observations and the number of observations should be at least 5 times as many variables. Comrey (1973) categorized the sample size for its suitability to factor analysis i.e., 100 as poor, 200 as fair, 300 as good, 500 as very good, and 1000 or more as excellent. Also, one can conduct Kaiser-Meyer-Olkin (KMO) test to check the sample adequacy. The sample is said to be adequate if KMO value is more than 0.5.

As far as correlation matrix is concerned, the observed variables should be linearly related but not highly correlated that may lead to the matrix as singular and create difficulty in determining the unique contribution of the variables to the factors. To check the correlation among variables, one can use Bartlett's test of spherity to test the null hypothesis that the correlation matrix is a identity matrix and the result should come out as significant. After rejecting the null hypothesis, one can validate the presence of multi-collinearity via the determinant of the correlation matrix i.e., if the determinant is greater than 0.00001, then there is no multi-collinearity (Field, 2000).

After getting correlation matrix, it is essential to determine whether factor analysis (FA) or principal component analysis (PCA) is to be performed. The main difference between these two lies on the way the eigen values are used. In PCA, all the diagonal elements of the correlation matrix are 1 and all the variance present in the dataset are accounted by the

components. However, in FA, the diagonal of the correlation matrix are squared multiple correlation coefficient, which is further used to get the eigen values and thereby the factor scores. Also, all the variances are not accounted by the factors as there is also an error variance. Further, in PCA the sum of square of the factor loadings of a variable provided the variance accounted for by that variable, which is not same in FA as it is assumed that the variables do not account for 100% of the variance. Theoretically, FA is more correct than PCA (Field, 2000) but practically there is little difference and is further decreased with decrease in the number of variables and increase in the value of factor loadings (Rietveld and Van Hout, 1993).

In conducting FA, one of the most important questions is the number of factors to be retained in the model. In PCA, the number of components is same as the number of positive eigen value. However eigen values are sometime positive and close to zero, and in that situation deciding the number of factor is difficult. In literature certain thumb rules are there to take decision about the number of factors. Guttman-Kaiser rule state that the factor with eigen value >1 should be retained in the model. Hair et al, (1995) stated that in the natural sciences the number factors retained in the model should explain at least 95% of the total variance present in the observed variables. In humanities, the number factors that can explain up to 60-70% variation may be retained in the model (Hair et al, 1995; Pett et al, 2003). Besides, another option is that first draw a scree plot (Cattell, 1966) and retained all those factors appeared before reaching the point of inflection.

After extracting the factors, the next task is to name the factors and interpret them. Since, most variable have higher value of loading on the most important factors and less amount of loadings on the remaining factors, it is always a difficult task to interpret about the factors. However, the factor rotation can help in this respect to a large extent. Factor rotation transforms the original loadings and thereby the interpretation becomes easier. Rotation maximizes the high loading items and minimizes the less loading items. There are two rotation techniques viz., orthogonal/ varimax and oblique/promax that are commonly used in factor analysis. Varimax rotation (Thomson, 2004) is the most common rotational technique used in factor analysis that produces uncorrelated factors. On the other hand, in oblique rotation, the factors are correlated. Often, the oblique rotation provides more accurate results when the data does not meet the prior assumptions. Further, to decide the type of rotation technique is almost difficult and therefore first carryout the analysis with oblique rotaions, and if the oblique rotation demonstrates a negligible correlation between the extracted factors then it is reasonable to use orthogonally rotated factors (Field, 2000). Regardless of the rotation techniques uses, the objective is to provide easier interpretation of the rrsults.

Interpretation of EFA is nothing but to determine which variables are attributed to a factor and labeling of that factor. However, the labeling of a factor is a subjective process (Henson

and Roberts, 2006), where the meaningful of the factor is dependent on the researchers definition. Moreover, through and systematic factor analysis is nothing but to find those factors that together explain the majority of the responses.

Mathematical aspects of EFA

Consider a dataset with *n* observations and *p* standardized variables $x_1, x_2, ..., x_p$. Then, in EFA the observed variables are expressed as the linear combination of the common factors and unique factor i.e., $x_i = a_{i1}F_1 + a_{i2}F_2 + a_{i3}F_3 + ... + a_{ik}F_k + e_i$, where i=1,2,…, p, k<p and a_{ik} is the factor loading of i^{th} variable on k^{th} factor which is not same as that of eigen vector. The assumptions of this model are $E(e_i) = 0$, $V(e_i) = \psi_i$, $E(e_i e_j) = 0$, $E(e_i F_j) = 0$ and $E(F_i F_j) = 0$. In matrix notation we can write $\mathbf{X}_{p\times n} = \mathbf{L}_{p\times k}\mathbf{F}_{k\times n} + \mathbf{E}_{p\times n}$, where

$$\mathbf{X}_{p\times n} = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1n} \\ x_{21} & x_{22} & x_{23} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ x_{p1} & x_{p2} & x_{p3} & \dots & x_{pn} \end{bmatrix}, \quad \mathbf{F}_{k\times n} = \begin{bmatrix} F_{11} & F_{12} & F_{13} & \dots & F_{1n} \\ F_{21} & F_{22} & F_{23} & \dots & F_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ F_{k1} & F_{k2} & F_{k3} & \dots & F_{kn} \end{bmatrix}, \quad \mathbf{L}_{p\times k} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1k} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2k} \\ \dots & \dots & \dots & \dots & \dots \\ a_{p1} & a_{p2} & a_{p3} & \dots & a_{pk} \end{bmatrix} \text{ and}$$

$$\mathbf{E}_{p\times n} = \begin{bmatrix} e_{11} & e_{12} & e_{13} & \dots & e_{1n} \\ e_{21} & e_{22} & e_{23} & \dots & e_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ e_{p1} & e_{p2} & e_{p3} & \dots & e_{pn} \end{bmatrix}.$$

Also, it is assumed that $E(\mathbf{E}) = 0$, $E(\mathbf{F}) = 0$, $\text{cov}(\mathbf{F},\mathbf{E}) = 0$, $V(\mathbf{E}) = Diag(\psi_1, \psi_2, ..., \psi_p) = \boldsymbol{\psi}(say)$ and $\text{var}(\mathbf{F}) = \mathbf{I}$. The correlation matrix is generally used for performing the factor analysis. Here the diagonal elements are 1 (often described as the variance of the observed variable). In PCA, this matrix is used as such but factor analysis involves the replacing of diagonal element with communality estimate. The communality estimate is the estimated proportion of variance of the variable that is free of error variance and is shared with other variables in the matrix. These estimates reflect the variance of a variable in common with all others together. The initial estimate of the communality is taken as the squared multiple correlation coefficients and then the communalities of the variables are estimated as the sum of the square of the loadings onto different factors. Once the correlation matrix of the observed variables are obtained, the factor analysis can be written as $\boldsymbol{\Sigma} = \mathbf{L}\mathbf{L}' + \boldsymbol{\psi}$, which nothing but $\text{var}(\mathbf{X}_{p\times n}) = \text{var}(\mathbf{L}_{p\times k}\mathbf{F}_{k\times n} + \mathbf{E}_{p\times n})$. So, for the i^{th} variable, one can write $1 = (a_{i1}^2 + a_{i2}^2 + ... + a_{ip}^2) + \psi_i$ or $1 = h_i^2 + \psi_i$ or Total variance=Variance explained by the common factors + Error variance. Here h_i^2 is the communality and 1-h_i^2 is the variance accounted for by the i^{th} unique factor. In this model, there is a need to estimate the common factor loadings (***L***) as well as the factor scores (***F***). For estimating ***L***, there are two methods

available one is Principal Axis Factor (PAF) method and other is Maximum Likelihood (ML) method. PAF makes no assumption about the error and minimizes the sum of squares of the residual matrix i.e., $\frac{1}{2}tr\left[(S-\Sigma)^2\right]=\sum_i\sum_j(s_{ij}-\sigma_{ij})^2$, where s_{ij} and σ_{ij} are the observed correlation matrix and implied correlation matrix, respectively (Jöreskog, 2007). The maximum likelihood (ML) estimation is derived from the theory of normal distribution. The ML value is obtained by minimizing $\ln|\Sigma|-\ln|S|+tr[S\Sigma^{-1}]-p$, which similar to minimizing the discrepancy function $\sum_i\sum_j\left[\frac{(s_{ij}-\sigma_{ij})^2}{\psi_i^2\psi_j^2}\right]$ (MacCallum et al, 2007).

For estimation of factor scores, generally three types of methods are used viz., ordinary least squares, weighted least squares and regression method. Let $\boldsymbol{x}_i$ be the i[th] observation vector and $\boldsymbol{f}_i$ is the corresponding vector of factor scores, then we can write $\mathbf{x}_i=\mathbf{L}\mathbf{f}_i+\mathbf{e}_i$, where i=1,2,.., *n*, and the estimates of factor scores for this model by different methods are provided as follows:

(I) **Ordinary Least Square**

The estimate of $\mathbf{f}_i$ can be obtained by minimizing the error sum of squares i.e., $\sum_{j=1}^{p}e_{ij}^2=\sum_{j=1}^{p}(x_{ij}-a_{i1}f_1-a_{i2}f_2-...-a_{ik}f)^2=(\mathbf{x}_i-\mathbf{L}\mathbf{f}_i)'(\mathbf{x}_i-\mathbf{L}\mathbf{f}_i)$. This is like a least squares regression, except in this case we already have estimates of the parameters (the factor loadings). In matrix notations, it can be written as $\hat{\mathbf{f}}_i=(\mathbf{L}'\mathbf{L})^{-1}\mathbf{L}'\mathbf{x}_i$. Using the principal component method with the unrotated factor loadings, the results can be obtained as

$$\hat{\mathbf{f}}_i=\begin{pmatrix}\frac{1}{\sqrt{\hat{\lambda}_1}}\hat{\xi}_1\mathbf{x}_i\\ \frac{1}{\sqrt{\hat{\lambda}_2}}\hat{\xi}_2\mathbf{x}_i\\ \cdots\\ \frac{1}{\sqrt{\hat{\lambda}_k}}\hat{\xi}_k\mathbf{x}_i\end{pmatrix},$$

where $\hat{\xi}_1,\hat{\xi}_2,\ldots,\hat{\xi}_k$ are the eigen vectors and $\hat{\lambda}_1,\hat{\lambda}_2,\ldots,\hat{\lambda}_k$ are the estimate of eigen values.

(II) **Weighted Least Squares**

In this method, larger weights are given to the variables having low specific variances. Variables with low specific variances are those for which the model fits the data best. In other words, the variable with the low specific variance provides more information regarding the true values for the specific factors. For the above considered model, we wish to minimize $\sum_{j=1}^{p}\frac{e_{ij}^2}{\psi_j}=\sum_{j=1}^{p}\frac{(x_{ij}-a_{i1}f_1-a_{i2}f_2-...-a_{ik}f)^2}{\psi_j}=(\mathbf{x}_i-\mathbf{L}\mathbf{f}_i)'\boldsymbol{\psi}^{-1}(\mathbf{x}_i-\mathbf{L}\mathbf{f}_i)$, that resulted in the estimate as $\hat{\mathbf{f}}_i=(\mathbf{L}'\boldsymbol{\psi}^{-1}\mathbf{L})^{-1}\mathbf{L}'\boldsymbol{\psi}^{-1}\mathbf{x}_i$. Both OLS and WLS methods are used for estimating the factor scores, while PAF method is used to estimate the factor loadings.

(III) **Regression method**

This method is used when maximum likelihood is used for estimating the factor loadings. Now, for standardized variables the joint distribution of $\mathbf{x}_i$ and $\mathbf{f}_i$ can be writes as $\begin{pmatrix}\mathbf{x}_i\\ \mathbf{f}_i\end{pmatrix}\sim N\left[\begin{pmatrix}0\\0\end{pmatrix},\begin{pmatrix}\mathbf{LL}'+\boldsymbol{\psi} & \mathbf{L}\\ \mathbf{L}' & \mathbf{I}\end{pmatrix}\right]$. Then, we can calculate the conditional expectation of the factor score $\mathbf{f}_i$ given the observed data $\mathbf{x}_i$ as $E(\mathbf{f}_i|\mathbf{x}_i)=\mathbf{L}'(\mathbf{LL}'+\boldsymbol{\psi})^{-1}\mathbf{x}_i$, which is nothing but the estimate of $\mathbf{f}_i$.

Step by step procedure for performing exploratory factor analysis using R

Step 1: Set the working directory. Let my directory is "meher" present in "D" drive. Then, set the directory as

setwd("C:/Documents and Settings/Prabin/Desktop/meher")

Step 2: Read the data from the specified directory. Let my data file is *fact.txt* present in the directory. Then data file can be imported to R as

x <- read.table (file= "fact.txt")

Step 3: Check the normality assumption of each variable using Shapiro-Wilk's test.

shapiro.test (x[,i]) # This is for i^{th} variable. If P-value is >level of significance, the variable is normally distributed.

Step 4: Check the adequacy of the each variable and sample as a whole for factor analysis using KSA and KMO and test. The desired value of KMO is > 0.5. Variables with MSA being below 0.5 indicate that item does not belong to a group and may be removed from the factor analysis.

kmo <- function(x)

{

x <- subset(x, complete.cases(x)) # Omit missing values

```
r <- cor(x)                                  # Correlation matrix
r2 <- r^2                          # Squared correlation coefficients
i <- solve(r)                       # Inverse matrix of correlation matrix
d <- diag(i)                        # Diagonal elements of inverse matrix
p2 <- (-i/sqrt(outer(d, d)))^2     # Squared partial correlation coefficients
diag(r2) <- diag(p2) <- 0          # Delete diagonal elements
KMO <- sum(r2)/(sum(r2)+sum(p2))
MSA <- colSums(r2)/(colSums(r2)+colSums(p2))
return(list(KMO=KMO, MSA=MSA))
}
kmo (x)
```

Step 5: Check that the correlation matrix is not an identity matrix using Bartlett's sphericity test. The test should come out significant.

```
bst <- function(x)
{
method <- "Bartlett's test of sphericity"
data.name <- deparse(substitute(x))
x <- subset(x, complete.cases(x)) # Omit missing values
n <- nrow(x)
p <- ncol(x)
chisq <- (1-n+(2*p+5)/6)*log(det(cor(x)))
df <- p*(p-1)/2
p.value <- pchisq(chisq, df, lower.tail=FALSE)
names(chisq) <- "X-squared"
names(df) <- "df"
return(structure(list(statistic=chisq, parameter=df, p.value=p.value,
```

method=method, data.name=data.name), class="htest"))

}

bst (x)

Step 6: Test that there is no presence of high degree of multicollinearity. The determinant of the matrix should come out > 0.0001 to pass the test.

det(cor(x))

Step 7: Carryout factor analysis to extract the factor loadings (by ML estimate method), common variances and specific variances.

factanal (x=swiss, factors=2, rotation= "varimax or promax")

or

factanal (~., factors=2, data=swiss, rotation= "varimax or promax")

In the result one cannot see the complete factor loadings but it is possible with the following commands.

factanal (~., factors=2, rotation= "varimax or promax")$loadings[,i] # for complete i^{th} factor loading.

Step 8: Estimate the factor scores either by Bartlett's WLS method or Johnson's regression method.

factanal (~., factors=2, rotation= "varimax or promax", scores="Bartlett or regression")$scores

Step 9: The factor loadings, common variances, specific variances can also be computed by supplying the covariance matrix and number of observations. However, the scores can only be obtained when full data set is available.

factanal (factors=2, covmat=cor(swiss),rotation= "varimax or promax", n.obs=47)

Step 10: Interpretation of the result and conclusion

Note: One can use the "psych" package of R-software for KMO test and Barlett's test of sphericity using single line code as provided below.

KMO(r) # r is the correlation matrix. This will provide the values of both KMO and KSA

cortest.bartlett(r, n) # r is the correlation matrix and n is the number of observation in the dataset.

Application of Random Forest in Genome Wide Association Studies

Decision tree learning is a method commonly used to create a model that predicts the value of target variable based on several input variables. Decision trees are of two types; (i) classification tree (ii) regression tree. Classification tree analysis is there when the response variable is a class label and the regression tree analysis is there when the response variable takes the values of real number. A classification tree is obtained by asking an ordered sequence of questions, where the type of questions asked at each step in the sequence depends upon the answers required for the previous questions of the sequence. The sequence always terminates in a prediction of the class label attached to the observation. The starting point of a classification tree is called the root node and consists of the whole data set at the top of the tree. A node in a tree can be a terminal or non-terminal node. A non-terminal (or parent) node is a node that split into daughter nodes. A node that doesn't split is called a terminal node and is assigned a class label. When an observation of unknown class is dropped down the tree and ends up at a terminal node, it is assigned to that class corresponding to the class label attached to that node. There may be more than one terminal node with the same class label. A single split tree with only two terminal nodes is called a stump. In case of binary splitted node, the split is determined by a Boolean condition on the value of a single variable, where the condition is either satisfied ("yes") or not satisfied ("no") by observed value of that variable. All the observations in the data set that have reached to a particular node and satisfy the condition for that variable drop down to one of the two daughter nodes and the remaining observations at that node that don't satisfy the condition drop down to the other daughter node.

Let x_1 and $x_{2\ be}$ two variables and θ_i (i=1, 2, 3, 4) be any values of the variables then the tree is grown by asking following questions:

(1) Is $x_2 \leq \theta_1$? If the answer is yes, follow the left branch;

if no follow the right branch.

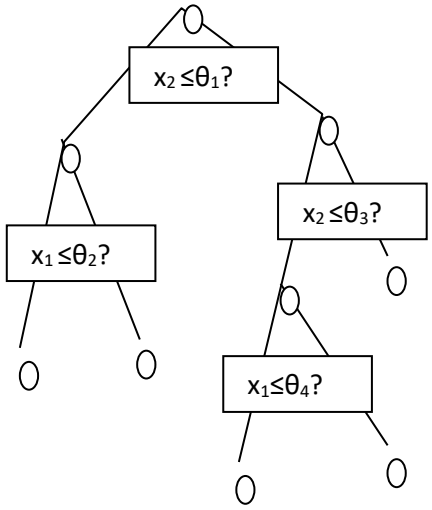
(2) If the answer to question (1) is yes, then ask the next question: Is $x_1 \leq \theta_2$?

if the answer is yes, follow the left branch (terminal);

if no follow the right branch (terminal).

(3) If the answer to question (1) is no, ask the next question: Is $x_2 \leq \theta_3$?

(4) if the answer to (3) is yes, {then ask the next question: Is $x_1 \leq \theta_4$?



if the answer is yes, follow the left branch (terminal);

if no follow the right branch (terminal)}.

if the answer to (3) is no, it leads to the terminal node

Aspect of growing Tree

For growing a classification tree, following four aspects need to be discussed

- Choosing the Boolean conditions for splitting at each node
- Criterion to be used to split a parent node into its daughter nodes
- To decide a node to become a terminal node
- Assigning a class to a terminal node

Splitting strategies

In the splitting strategy the first two aspects of growing tree are discussed.

Number of possible splits

For continuous or ordinal variable, the total number of possible splits at a given node is one fewer than the number of its distinctly observed values. For nominal or categorical variable of *m* distinct categories, there will be 2^{m-1}-1dinstict splits at a particular node.

Node impurity function

To chose the best split among all variables, first chose the best split for a given variables by using measure of goodness of split. Let $\Pi_1,\ldots,\Pi_K$ be the K≥2 classes. For node τ, the node

impurity function i(τ) is given as $i(\tau) = \phi(p(1|\tau),...,p(K|\tau))$,Where $p(k/\tau)$ is an estimate of $P(Đ_K/ô)$ which is the conditional probability that an observation **X** is in $Đ_k$ given that it falls into

node ô. The function ϕ will attain maxima at the point $(\frac{1}{K},\frac{1}{K},\ldots,\frac{1}{K})$ on the set of K-tuples of probabilities $(p_1,\ldots,p_K)$ and its sum is unity. In the two classes case (K=2), these condition reduces to a symmetric $\phi(p)$ maximized at the point p=1/2. One such function is the entropy function,

$$i(\tau) = -\sum_{k=1}^{K} p(k|\tau)\ \log p(k|\tau)$$

and for binary classes it reduces to $i(\tau) = -p\ \log p - (1-p)\ \log(1-p)$

Choosing best split for a variable

Let at node ô, after applying split *s*, a portion p_l goes to the daughter node $ô_l$ and the remaining portion p_r goes to the right daughter node $ô_r$. Then the goodness of split *s* at nod ô is the reduction in impurity gained by splitting the parent node ô in to its daughter nodes $ô_l$ and $ô_r$, which is given by .

$$\Delta i(\tau) = i(\tau) - p_l\, i(\tau_l) - p_r\, i(\tau_r)$$

For example, consider a data set having the response variable y that has two values 0 and 1 and suppose one of the possible split of the input variables x_j is $x_j \leq c$ vs. $x_j > c$, where c is some values of x_j. Then a 2×2 table can be prepared as follows:

	1	0	Row total
$x_j \leq c$	n_{11}	n_{12}	$n_{1.}$
$x_j > c$	n_{21}	n_{22}	$n_{2.}$
Column Total	$n_{.1}$	$n_{.2}$	$n_{..}$

Now for the parent node ô, $p_l=(n_{.1} / n_{..})$ and $p_r=(n_{.2} / n_{..})$ so the impurity function at the parent node will be

$$i(\tau) = -\left(\frac{n_{.1}}{n_{..}}\right)\log_e\left(\frac{n_{.1}}{n_{..}}\right) - \left(\frac{n_{.2}}{n_{..}}\right)\log_e\left(\frac{n_{.2}}{n_{..}}\right)$$

Now for the daughter nodes $ô_l$ and $ô_r$, for $x_j \leq c$, $p_l=(n_{11} / n_{1.})$ and $p_r=(n_{12} / n_{1.})$ and for $x_j > c$, $p_l=(n_{21} / n_{2.})$ and $p_r=(n_{22} / n_{2.})$. Then the impurity function at the daughter nodes will be

$$i(\tau_l) = -\left(\frac{n_{11}}{n_{1.}}\right).\log_e\left(\frac{n_{11}}{n_{1.}}\right) - \left(\frac{n_{12}}{n_{1.}}\right).\log_e\left(\frac{n_{12}}{n_{1.}}\right)$$

$$i(\tau_r) = -\left(\frac{n_{21}}{n_{2.}}\right).\log_e\left(\frac{n_{21}}{n_{2.}}\right) - \left(\frac{n_{22}}{n_{2.}}\right).\log_e\left(\frac{n_{22}}{n_{2.}}\right)$$

and the best split for the single variable x_j is the one that has largest value of $\Delta i(s,\tau)$ over all *s* ^{a} S_j, the set of all possible split for x_j .

Choosing best split at a node

A tree starts with the root node, which consists of all observation. By using the goodness-of-fit criterion for a single variable, the best split at the root node for each of the variables x_1 to x_r can be found. The best split *s* at the root node is then the one that has the largest value of $\Delta i(s,\tau)$ over all *r* single-variable best splits at that node.

Choosing terminal node

A node can be declared as a terminal node if it fails to be larger than certain predetermined size; that is , if $n(\hat{o}) \leq n_{min}$, where $n(\hat{o})$ is the number of observations in node $\hat{o}$ and n_{min} is some previously assumed minimum size of a node. The terminal node act as a break on the tree growth, the larger the value of n_{min}, the more severe the break. In another way a node can be declared as a terminal node if the largest goodness-of-fit value at that node is smaller than a certain predetermined limit. However, these stooping rules are not fruitful in reality. A better approach is to let the tree grow to saturation and then prune it back (Breiman *et al*. 1984).

Associating a class with the terminal node

Suppose at a terminal node $\hat{o}$ there are $n(\hat{o})$ observation of which $n_k(\hat{o})$ are from class Π_k, k=1,…,K. then the class which corresponds to the largest of the $\{n_k(\hat{o})\}$ is assigned to $\hat{o}$. This is called plurality rule and it can be easily obtained from the Bayes's rule classifier, where the node $\hat{o}$ can be assigned to the class Π_i if

$$p(\Pi_i \mid \tau) = \max_{1 \leq k \leq K} p(\Pi_k \mid \tau)$$

Let $p(\hat{o}^a \Pi_i) = p_i$, (i=1,…,K), be the prior probability of the nod $\hat{o}$ belonging to different classes i.e., $p_i = n_i(\hat{o})|n(\hat{o}))$ and let $p_i(\hat{o}) = p(\hat{o}|\Pi_i)$ be the probability distribution function of observations in node $\hat{o}$ belonging to class Π_i. then the posterior probability of that node $\hat{o}$ will be assigned the class Π_i is given by

$$p(\Pi_i \mid \tau) = \frac{p_i(\tau).p_i}{\sum_{k=1}^{K} p_k(\tau).p_k}$$

The Bayes's rule classifier for K classes assigns $\hat{o}$ to that class with the highest posterior probability. Since the denominator is fixed for all the classes, the node $\hat{o}$ will be assigned to the class Π_i if

$$p(\Pi_i \mid \tau) = \max_{1 \leq k \leq K} p(\Pi_k \mid \tau)$$

Ensemble of classifiers

A well-known method of building classification systems is to build multiple classifiers, each from a subset of the original training set, such that the final classification decision is aggregated from all classifiers' decisions. This method is called the *classifier ensemble* method (Buhlmann *et al.* 2004). For example, five classifiers could be built independently using five different subsets of the original training set. These five classifiers would produce five predictions of the class label for each new record, and the class with a plurality of votes would be the prediction of the entire ensemble. It is also possible to extend this simple voting

scheme so that each individual classifier prediction is given a weight, perhaps based on its test accuracy. The overall prediction becomes the plurality of the weighted votes.

Classifiers in an ensemble can all have the same type, or they can be of different types. For example, an ensemble with three classifiers can consist of three decision trees, or it can consist of a decision tree, a neural network (Kantardzic, 2003), and a Bayesian network (Dunham, 2003). Both kinds of ensembles are known to perform better than single classifiers. The variance between classifiers is reduced in the case of classifiers of the same type, and the bias between classifiers is reduced for ensembles with different types of classifiers. The classification models of ensembles for both kinds are, therefore, more representative of the data than a single classifier. In other words, having multiple strong classifiers each built from a different sample of the dataset leads to a final classification decision with higher accuracy than a single classifier.

Generating the datasets used for training the classifiers in an ensemble can be done by different methods such as bootstrap sampling (bagging) (Breiman, 1994), and boosting (Freund and Schapire, 1996). Suppose that a dataset contains n records, each with m attributes. Bootstrap sampling or bagging generates the datasets each of size n by randomly sampling the records with replacement. Hence the training dataset for each tree contains multiple copies of some of the original records. Boosting maintains weights for records in the training set, such that these weights are updated after each classifier is trained according to the difficulty of classifying the current set of records. The weights are then used to derive the new sampling for the dataset.

Random Forest

Bagging (Bootstrap aggregating) was the first procedure that successfully combined the ensemble of tree classifiers to improve the performance over a single classifier (Breiman, 1996b). In bagging randomization is introduced only while selecting the data set on which each tree is grown. Random forest (Breiman, 2001) is an extension of this bagging procedure where another source of randomization is introduced by choosing a subset of m variables at each node and node is split on the basis of best split.

Let $L = \{(\mathbf{x}_i, y_i), i = 1, 2, ..., n\}$ is the learning data set where y_i is the response variable and it takes values from K classes and there are p variables in the data set. Random forest consists of ensemble of B classifiers $h_1(\mathbf{x}), h_1(\mathbf{x}), ..., h_B(\mathbf{x})$, where each classifier is constructed upon a bootstrap replica of the learning data set, by selecting randomly selecting a subset of m variables out of p variables and the best split is determined on the basis of m selected variables using gini index.. Each classifier votes for one of the classes for each test instances and test instance is classified by the label of winning class. As the individual trees are

constructed upon a bootstrap replication, there is on an average 36.8% of instances are not playing any role in the construction of the tree. These instances are called out of bag (OOB) instances. These OOB instances are the source of data used in the random forest for estimating the classification error and to evaluate the performance of the random forest. Random forests are computationally very efficient and offer good prediction accuracy and are less sensitive to noisy data.

Some features of RF

Let (**x**, y) denote the learning instances having n number of observations where each vector of attributes **x** is labeled with class y_j, (j=1,2,…,c). The correct class is denoted by y. $p(y_j)$ is the probability of class y_j . denote the set of OOB instances for classifier h_b as O_b. Let $Q(\mathbf{x}, y_j)$ be the OOB proportion of votes for class y_j for input vector **x.**

$$Q(\mathbf{x}, y_j) = \frac{\sum_{b=1}^{B} I(h_b(\mathbf{x}) = y_j; (\mathbf{x}, y) \in O_b)}{\sum_{b=1}^{B} I(h_b(\mathbf{x}); (\mathbf{x}, y) \in O_b)}$$

The Margin function, strength and Correlation between classifiers in a RF is defined as follow.

Margin function- The "margin function" measures the extent to which the average vote for right class y exceeds the average vote for any other class. The margin function of the labeled observation (**x,** y) is $m(\mathbf{x}, y) = P(h(\mathbf{x}) = y) - \max_{\substack{j=1 \\ j \neq y}}^{c} P(h(\mathbf{x}) = y_j)$. If m(x, y)>0, then h(**x**) correctly classifies y. h(**x**) denote a classifier that predict the label y for an observation **x.**

Strength- It is defined as the expected margin, and is computed as the average over the training set.

$$s = \frac{1}{n}\sum_{i=1}^{n}\left(Q(\mathbf{x}_i, y) - \max_{\substack{j=1 \\ j \neq y}}^{c} Q(\mathbf{x}_i, y_j) \right), where$$

$$Q(\mathbf{x}_i, y_j) = \frac{\sum_{b=1}^{B} I(h_b(\mathbf{x}) = y_j; (\mathbf{x}, y) \in O_b)}{\sum_{b=1}^{B} I(h_b(\mathbf{x}); (\mathbf{x}, y) \in O_b)}$$

$$Q(\mathbf{x}_i, y) = \frac{\sum_{b=1}^{B} I(h_b(\mathbf{x}) = y; (\mathbf{x}, y) \in O_b)}{\sum_{b=1}^{B} I(h_b(\mathbf{x}); (\mathbf{x}, y) \in O_b)}$$

where I(.) is the indicator function

Advantages

- People could understand and interpret easily after brief explanation
- Many data analysis techniques require data normalization, creation of dummy variable etc. but it requires little data preparation.
- Generally the techniques are specialized in analyzing data set having only one type of variable, but it handles both numerical and categorical data.
- Performs well with large data in a short time

SNP Detection

Single Nucleotide polymorphism (SNPs): - SNPs are the variations in individual's building blocks (base pairs) of DNA sequences that are distributed randomly over the genome and passed from generation to generation. Identification of SNPs is important in several applications of Microarrays, including Genotyping, forensic analysis, identification of disease, identifying drug-candidates, evaluating germline mutations in individuals, assessing loss of heterozygosity, or genetic linkage analysis and many more.

In CLC Genomic Workbench, the SNP detection will scan through the entire data for the SNPs.

Toolbox | High-throughput Sequencing

↓

Roche 454| SNP detection

An E. coli data set consisting of a little more than 400,000 reads from a 454 sequencer is used here as an example data set. Select the Ecoli.FLX.fna and Ecoli.FLX.qual with the Remove adapter sequence checkbox is checked and that the paired-end reads checkbox is NOT checked.

Next

↓

Finish and save

↓

After a short while, the reads have been imported

↓

Next, import the reference genome sequence (e.g. NC_010473.gbk)

Mapping the reads to reference

The first step is to map the reads to the reference genome.

Select the E.coli.FLX sequence list

↓

Toolbox| High-throughput Sequencing

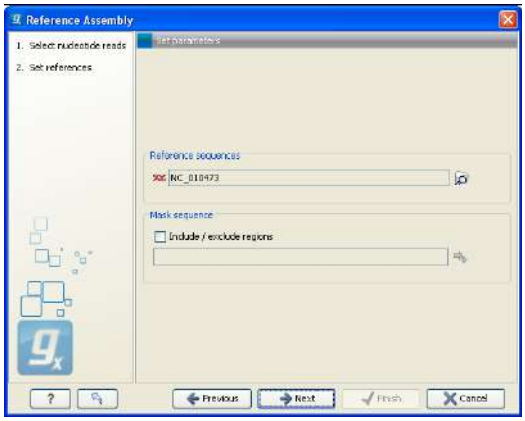
↓

Reference Assembly

Select sequence list containing the reads. The reference sequence will be selected in the next step.

↓

Next, import the reference genome sequence (e.g. NC_010473.gbk)



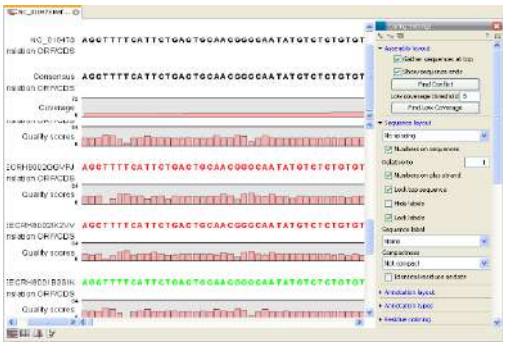
Specifying the reference sequences and masking.

Results of the reference assembly

List of non-assembled reads: It will give the list of reads that did not match the reference sequence. This list can be used to investigate the contamination in the sample or structural differences between the sequencing data and the reference sequence followed by performing de novo assembly of these reads and then use BLAST to investigate the contigs.

Report: This report shows the information about the assembly and the number of reads that matched the reference sequence.

Contig: The contig shows the alignment of all the reads to the reference including quality scores. For annotated reference sequences it displays the translation of the coding regions (the yellow CDS annotations) in the Side Panel in the Nucleotide info group under Translation.



The contig with the reads mapped to the reference.

Parameters

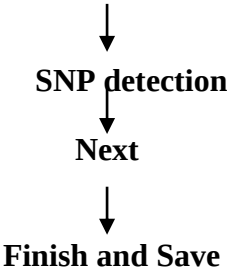
Sequence layout –Compactness: It is used to used to determine the height of the reads

Alignment info – Coverage: It displays a graph of the coverage along the contig. The read colors are green (forward) and red (reverse) by default.

Looking for SNPs

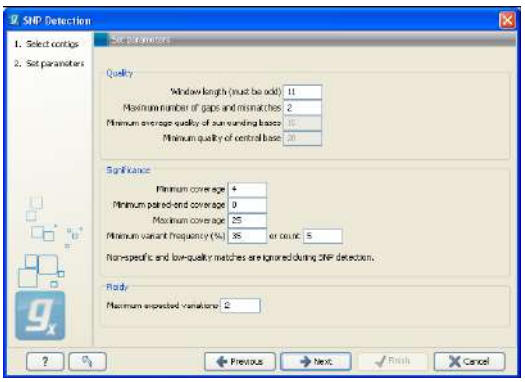
CLC Genomics Workbench provides two tools: SNP and DIP detection to help you get an overview of the differences between the reference sequence and the reads.

Toolbox | High-throughput Sequencing



Parameters

Add SNP annotations to reference: This will add an annotation for each SNP to the reference sequence.

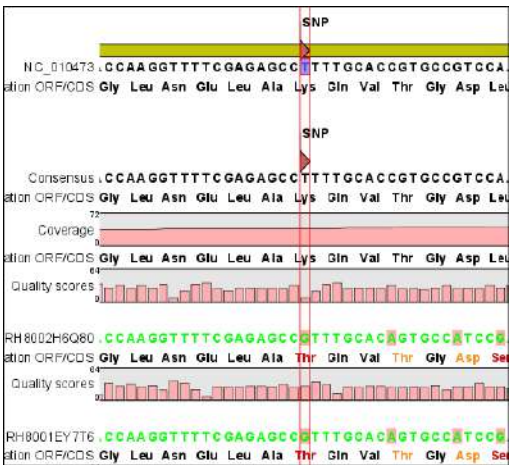


SNP detection parameter

Add SNP annotations to consensus: This will add an annotation for each SNP to the consensus sequence.

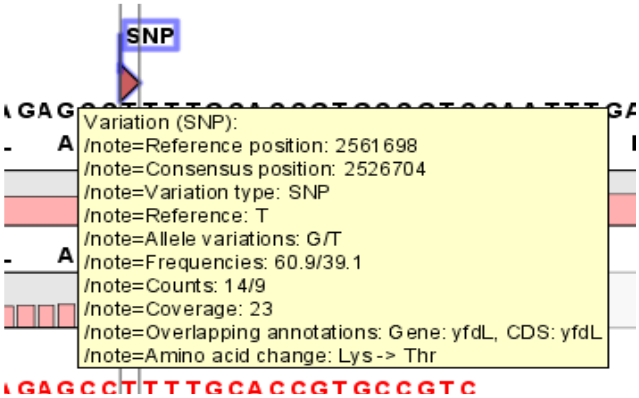
Create table: This will create a table showing all the SNPs found in the data set with all the valuable overview, whereas the annotations are useful for detailed inspection of a SNP, and also if the consensus sequence is used in further analysis.

SNP Annotation



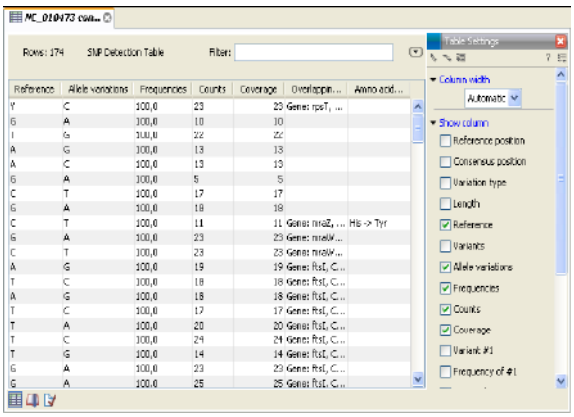
A SNP annotation within a coding region

The SNP in the above figure is showing a coding region and with one of the variations actually changes the protein product (from Lys to Thr).

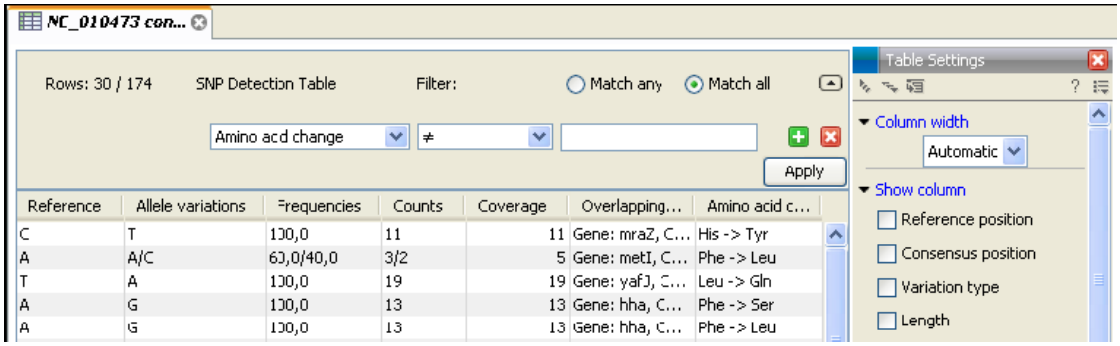


A SNP annotation with associated information

The result of the SNP detection will now open a table



A table of SNPs.



Filtering the SNP table to only display nonsynonymous SNPs.

References

Breiman, L., Friedman, J., Olshen, R. and Stone, C. (1984). *Classification and regression trees*, Boca Raton, FL: Wadsworth.

Cattell, RB (1966) The scree test for the number of factors. *Multivariate Behavioral Research*, 1, 245-276.

Comrey, AL (1973) *A First Course in Factor Analysis*. New York: Academic Press, Inc.

Field, A (2000) *Discovering Statistics using SPSS for Windows*. London – Thousand Oaks – New Delhi: Sage publications.

Habing, B (2003) *Exploratory Factor Analysis*. Website: http://www.stat.sc.edu/~habing/courses/530EFA.pdf

Hair, J., Anderson, RE., Tatham, RL., Black, WC (1995) *Multivariate data analysis*. 4th edn. New Jersey: Prentice-Hall Inc.

Henson, RK., Roberts, JK (2006) Use of Exploratory Factor Analysis in Published Research: Common Errors and Some Comment on Improved Practice. *Educational and Psychological Measurement*, 66(3).

Izenman, A. J. (2008). *Modern multivariate statistical techniques; regression, classification and manifold learning*, New York: Springer. Johnson, R.A. and Wichern, D.W. (1996). Applied multivariate statistical analysis. *Prentice-Hall of India Private Limited*.

Jöreskog, G (2007) *Factor analysis and its extensions*, in *Factor analysis at 100: Historical Developments and Future Directions*, R. Cudeck and R.C. MacCallum, eds., Lawrence Erlbaum, Mahwah, NJ, pp. 47–77.

Kass, G. V. (1980). An explanatory technique for investigating large quantities of categorical data, *Applied Statistics*, **29**, 119-127.

MacCallum, RC., Browne, MW., Cai, L (2007) *Factor analysis models as approximations*, in *Factor Analysis at 100: Historical Developments and Future Directions*, R. Cudeck and R.C. MacCallum eds., Lawrence Erlbaum, Mahwah, NJ, pp. 153–175.

Pett, MA., Lackey, NR., Sullivan, JJ (2003) *Making Sense of Factor Analysis: The use of factor analysis for instrument development in health care research*. California: Sage Publications Inc.

Rietveld, T., Van Hout, R (1993) *Statistical Techniques for the Study of Language and Language Behaviour*. Berlin – New York: Mouton de Gruyter.

Thompson, B (2004) *Exploratory and confirmatory factor analysis: understanding concepts and applications*. Washington, DC: American Psychological Association.

Zhang, H. and Singer, B. (1999). Recursive portioning in the health sciences, New York: Springer.

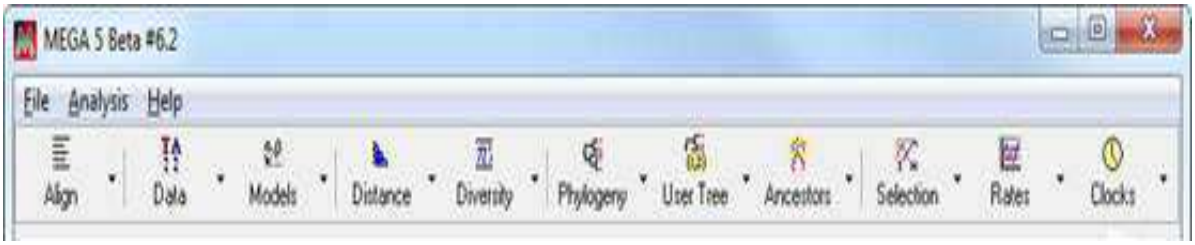
Phylogenetic Analysis

(Practical)

Before we can perform any kind of analysis, we have to activate a MEGA file. A Mega file is file that contains a multiple sequence alignment that has been exported from the Alignment Explorer in Mega file format. We can either activate a Mega file that we have saved somewhere on our computer or flash drive, or we can create a new Mega file from a creating a multiple sequence alignment in the Alignment Explorer and then exporting the alignment as a Mega file.

Creating Multiple Sequence Alignments with Alignment Explorer

I) Creating multiple sequence alignment from an open text file



1. Launch the Alignment Explorer by selecting **Alignment -> Alignment/CLUSTAL**
2. A window will appear asking you either to a) Create a new alignment, b) Open a saved alignment session, or c) Retrieve sequences from a file. Select the first option, "create a new alignment".
3. Copy and paste unaligned sequences from the text file to the Alignment Explorer.
4. In the Alignment Explorer highlight all the sequences by selecting **Edit -> Select All.**
5. Align the highlighted sequences by selecting **Alignment -> Align by ClustalW.**
6. Save the current alignment as an alignment session file by selecting **Data -> Export -> Save.** This will allow the current alignment session to be restored for future editing in a file with the extension ".mas", i.e. cox_alignment.mas
7. Save the current alignment as a MEGA file by selecting **Data -> Export -> MEGA file.** This will allow the current alignment to be analyzed by MEGA.

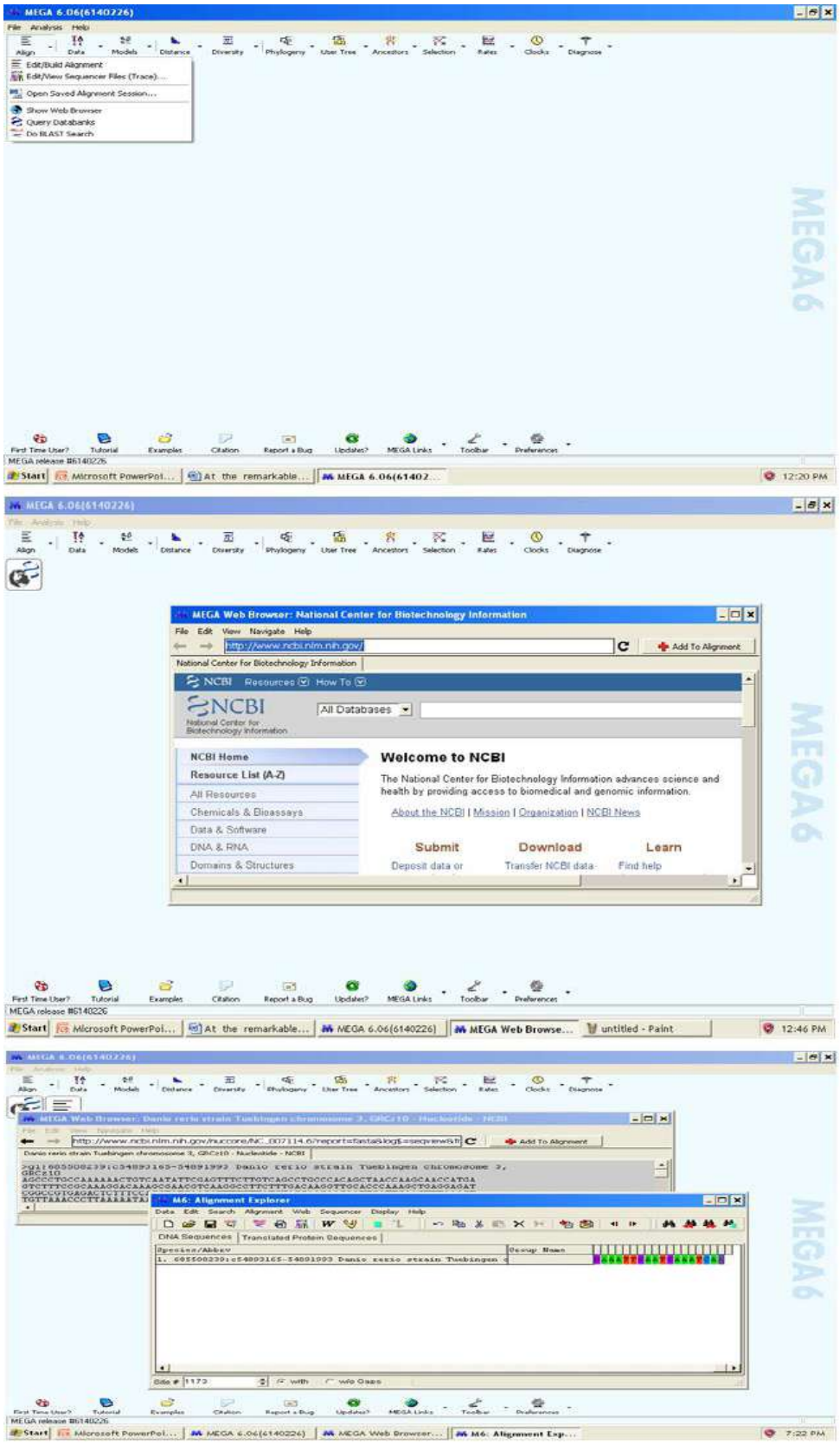


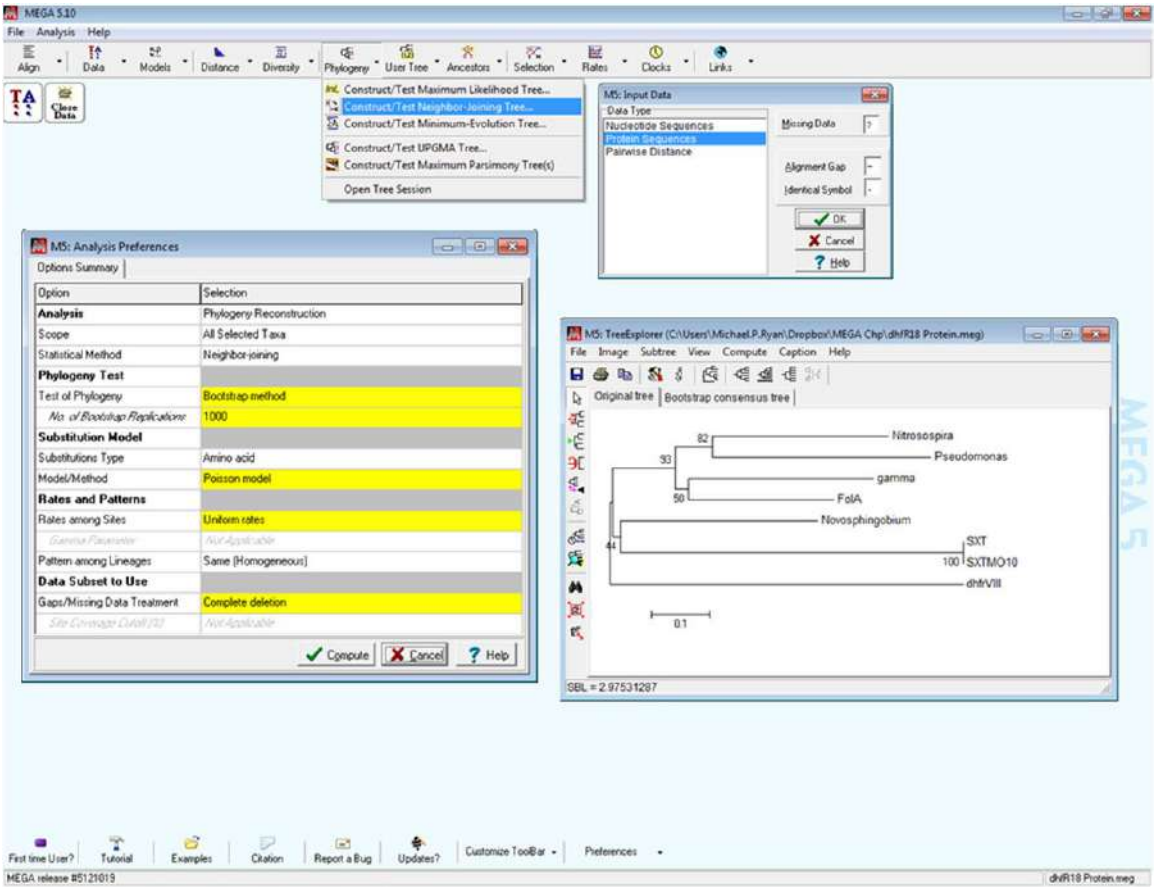
Figure 1: sequence retrieval in MEGA software

II) ***Creating multiple sequence alignment from the MEGA web browser using text searching***

1. Activate the Web Explorer tab by selecting **Alignment -> Show Web Browser.**
2. When NCBI is loaded, select any one of the databases, i.e. Entrez Gene, nucleotide, or protein. Enter your search terms for your desired sequence.
3. When the search results are displayed click the box to the left of each accession number whose sequence information you would like to download
4. Change the display of your results in the browser to fasta format by changing the selection in the Display drop box **Summary** to **Fasta**.
5. Change the screen from html format to text format by changing the drop box with **Send to** selected to **Text.**
6. To add these sequences to the Alignment Explorer, click the **Add to Alignment** button and these sequences will be automatically added. Steps 1 through 6 can be repreated several times and more sequences will continually be added to the Alignment Explorer.
7. Align the sequences using the steps detailed in above beginning at step 4.

III) Creating multiple sequence alignment from the MEGA web browser using BLAST

1. Activate the Web Explorer tab by selecting **Alignment -> Show Web Browser.**
2. Click on the **BLAST** hyper link on the NCBI homepage
3. Select the appropriate BLAST program for your search by clicking on one of the 5 options.
4. Copy in your sequence and perform your blast search.
5. When your results are returned, lcick on the box to the left of each accession number whose sequence information you would like to download. (*If you want to select everything, click the* ***Select All*** *button*).
6. Now follow steps in the previous section beginning at step 4.



Constructing Trees

1. Activate the data file that you want to analyze by clicking the link **Click me to activate a data file**.

2. Select the **Phylogeny -> Construct Tree -> Neighbor-Joining** command to display the analysis preferences dialog box.

3. In the Options Summary tab, click the Model pulldown (found in the Substitution Model section) and then select the **Amino Acid -> p-distance option**. A progress indicator will appear briefly, then the tree will be displayed in the Tree Explorer.

4. To select a branch, click on it with the left mouse button. IF you click on a branch with the right mouse button, you will get a small options menu that will let you flip the branch and perform various other operations on it. To edit the OUT labels, double click on them.

5. Change the branch style by selecting the **View->Tree/Branch Style** command from the Tree Explorer menu.

6. At this time the cursor assumes a triangular shape instead of the diamond shape. Press M and the mirror image of the original tree is displayed instantly. Press M again and the tree reverts to its original shape.

7. Select the **View -> Topology Only** command from the Tree Explorer menu and the branching pattern (without actual branch lengths) is displayed on the screen. Press T again and the actual NJ tree reappears.

8. Press F1 to examine the help for tree editor. Use the help to become familiar with the many operations that Tree Explorer is capable of performing.

9. DO NOT remove the tree from the screen. We shall use it for illustrating how a tree can be printed.

Printing the NJ tree

1. Select the **File –> Print** command from the Tree Explorer menu to bring up a standard Window print dialog.

2. To restrict the size of the printed tree to a single sheet of paper, choose the **File->Print in a Sheet** command from the Tree Explorer menu.

Constructing a maximum parsimony tree by using the branch-&-bound *search option.*

1. Select **Phylogeny -> Construct Tree -> Maximum Parsimony** command. In the resultant preferences window, choose the Max-Mini Branch-&-Bound Search option in the MP Tree Search Options tab.

2. Click the "OK" button to accept the defaults for the other options and begin the calculation. A progress window will appear briefly and the tree will be displayed in Tree Explorer.

3. Now print this tree.

4. Exit out the tree.

5. Compare the NJ and MP trees.

Test the reliability of a Tree Obtained.

1. Select the **Phylogeny -> Bootstrap Test of Phylogeny->Neighbor-Joining** from the main application menu.

2. An analysis preferences dialog box appears. Use the Models pulldown to ensure that **Amino-Acid -> p-distance** model is selected. Note that only the Amino-Acid submenu is available.

3. Click “Compute” to accept the default values for the rest of the options and compute the tree.

4. Once the computation is complete, the Tree Explorer appears and display two tree tabs. The first is the original Neighbor-Joining tree and the second is the Bootstrap consensus tree.

5. To produce a condensed tree, use the **Compute -> Condensed Tree** menu command from the Tree Explorer menu. This tree shows all the branches that are supported at the default cutoff value of **BCL >= 50**. To change this value, select the **View->Options** menu command and click the cutoff values tab. Select the **Compute ->Condensed Tree** menu command and the NJ tree will reappear.

6. Print this tree

Genome Wide Association Study (GWAS)

A genome-wide association study is an approach that involves rapidly scanning markers across the complete sets of DNA, or genomes, of many subjects to find genetic variations associated with a particular trait. Once new genetic associations are identified, researchers can use the information to develop better strategies to detect and manage the trait. Such studies are particularly useful in finding genetic variations that contribute to common, complex traits. In other words, Genome Wide Association Studies (GWAS) is based on correlations between genetic markers (usually Single Nucleotide Polymorphisms, short SNPs) and any measurable trait in a population of individuals. The main motivation in identifications of these associations is to find out new candidates for causal variants in genes (or their regulatory elements) that play a role for the phenotype of interest. This may eventually lead to a better understanding of the genetic components of the trait. Current GWAS usually include the following steps:

- Genotype calling from the raw chip-data and basic quality control.
- Principle Component Analysis (PCA) to detect and possibly correct for population stratification.
- Genotype imputation (using linkage disequilibrium information from HapMap).
- Testing for association between a single SNP and continuous or categorical phenotypes.
- Global significance analysis and correction for multiple testing.
- Data presentation (e.g. using quantile-quantile and Manhattan plots).
- Cross-replication and meta-analysis for integration of association data from multiple studies.

It has been found that (meta-) studies with many thousands (and even ten-thousands) of samples could at best identify a few (dozen) candidate loci with highly significant associations. Although, these unknown associations have been replicated in independent studies, each locus explains but a tiny (<1%) fraction of the genetic variance of the phenotype. Number of reasons could be attributed to this fact. Some important reasons are as follows:

- Estimation of heritability of trait from one generation to another is a problem especially for low heritable traits.
- Often genotype information is incomplete. For example, most analyses used microarrays probing of fractions of SNPs, while many of these SNPs can be imputed accurately using information on linkage disequilibrium. There still remains a significant fraction of SNPs which are poorly tagged by the measured SNPs. Furthermore, rare

variants with a Minor Allele Frequency (MAF) of less than 1% are not accessed at all with SNP-chips, which may nevertheless be the causal agents for many phenotypes. Finally, other genetic variants like Copy Number Variations (CNVs) (or even epigenetics) may also play an important role.

- Current analyses usually only employ additive models considering one SNP at a time with few co-variables and principle components reflecting population sub-structures. This obviously covers a small set of all possible interactions between genetic variants and the environment. Even more challenging task is taking into account purely genetic interactions, since already the number of all possible pair-wise interactions scales like the number of genetic markers squared.

Micro satellites markers are generally used for finding association with a candidate gene or linked region of a chromosomes. This is due to the fact that linkage exists over a very broad region and entire chromosome can be divided only 400-800 DNA markers regions. This can be used for population/family based designs. Using SNPs are more appropriate in other cases but cost plays an important role in this case.

Single Nucleotide Polymorphisms

The modern unit of genetic variation is the Single Nucleotide Polymorphism or SNP. SNPs are single base-pair changes in the DNA sequence that occur with high frequency in a genome. For the purposes of genetic studies, SNPs are typically used as markers of a genomic region, with the large majority of them having hardly any impact on biological systems. SNPs can have functional consequences, however, causing amino acid changes, changes to mRNA transcript stability, and changes to transcription factor binding affinity. SNPs are by far the most abundant form of genetic variation in living organism. SNPs typically have two alleles, meaning within a population there are two commonly occurring base-pair possibilities for a SNP location. The frequency of a SNP is given in terms of the minor allele frequency or the frequency of the less common allele. Rare SNPs i.e. SNPs with low frequency in the population are sometimes referred to as mutations though they can be structurally equivalent SNPs - single base-pair changes in the DNA sequence. In the genetics literature, the term SNP is generally applied common single base-pair changes, and the term mutation is applied to rare genetic variants. In is well known fact that common traits are likely to be influenced by genetic variation that is also common in the population. Also, if common genetic variants influence the trait, the effect size for any one variant must be small relative to that found in rare trait. Therefore, the allele frequency and the population prevalence are completely correlated. However, if a SNP caused a small change in gene expression that has small effect, then the influential allele would be only slightly correlated. Further, if common alleles have small genetic effects, but show heritability then multiple common alleles must influence disease susceptibility. These points suggest that traditional family-based genetic studies are not likely to be successful for complex

traits, prompting a shift toward population-based studies. The frequency with which an allele occurs in the population and the risk incurred by that allele for complex trait are key components to consider when planning a genetic study along with impact of the technology needed to gather genetic information and the sample size needed to discover statistically significant genetic effects. Under these circumstances we need to go for GWAS. GWAS needs large sample sizes and a large panel of genetic markers technology to gather genetic information to discover statistically significant genetic effects.

Linkage disequilibrium (LD) mapping of QTL exploits population level associations between markers and QTL. These associations arise because there are small segments of chromosome in the current population which are descended from the same common ancestor. These chromosome segments, which trace back to the same common ancestor without intervening recombination, will carry identical marker alleles or marker haplotypes, and if there is a QTL somewhere within the chromosome segment, they will also carry identical QTL alleles. There are number of QTL mapping strategies which exploit LD, the simplest of these is the genome wide association test using single marker regression

Genome Wide Association Study

It refers to a method / methodology for interrogating large number of variable points across a genome. As these variations are inherited in groups, or blocks, not all points have to be tested. It is an approach which involves rapidly scanning markers across the complete sets of DNA, or genomes of number of subjects to find genetic variations associated with a particular trait. Once new genetic associations are identified, researchers can use this information to develop better management strategies. Genome-wide association studies were made possible by the availability of chip based microarray technology for assaying one million or more SNPs. Two primary platforms have been used for most GWAS i.e. Illumina and Affymetrix. The Affymetrix platform prints short DNA sequences as a spot on the chip that recognizes a specific SNP allele. Alleles (i.e. nucleotides) are detected by differential hybridization of the sample DNA. Illumina on the other hand uses a bead-based technology with slightly longer DNA sequences to detect alleles. The Illumina chips are more expensive to make but provide better specificity. It is important to note that the technology for measuring genomic variation changing rapidly. Chip-based genotyping platforms are being replaced over the years with inexpensive new technologies for sequencing the entire genome i.e. next-generation sequencing methods. There are two primary classes of phenotypes categorical i.e. binary (case/control) and quantitative. Statistically, quantitative traits are preferred because as they improve power to detect a genetic effect, and often have a more interpretable outcome. The study design for this genetic association differs based on (i) scale of study i.e. genome wide based or genomics based, (ii) marker design, which depends on selection of best marker i.e. microsatellite, SNP and CNV (iii) subject design i.e. based on candidate gene or genome wide screening approach.

The genome wide studies mainly classified in to three categories i.e. cohort studies, family based study and case-control studies. In case of cohort studies, the subjects are assumed to be representative of the population. The phenotypes are used to ascertain the similarity among these subjects irrespective of genetic variations. This technique directly measures the risk and also less biased then case control studies. But it requires long follow up with large sample size. It also very expensive and poorly suited for rare traits. In case of family based studies, the basic assumption is that families are representative of the population of interest and both parents are from same genetic background. The major advantage of this technique is that, it checks for Mendelian in heritance and less prone to spurious associations. In this case, parent phenotypes are not required. It also allows for investigation imprinting and simple logistics techniques are applicable to detect the association. It is cost inefficient, with low power and very sensitive to genotyping errors. Third types of studies are known as case-control studies. In these studies, subjects are drawn from same population and cases represent all cases of the population. These are simple, cheap, and we can use large number of case and control variables. These are optimal for studying rare traits. In this, results are prone to population stratification criterion. In this, batch effect and other biases plays a major role. Generally it gives over estimation for common traits. Mostly, GWAS are used in diseased studies. In case of disease studies, there are three types of diseases as follows:

- Monogenic diseases: This is also a single gene produce disease. Often these disease are severe and appear early in life cycle. For the population as a whole, they are relatively rare. In a sense, these are pure genetic diseases. They do not require any environmental factors to elicit them. Although, nutrition is not involved in the causation of monogenic diseases, these diseases can have implications for nutrition. They reveal the effects of particular proteins or enzymes that also are influenced by nutritional factors.
- Oligogenic diseases: These are conditions produced by the combination of two, three, or four defective genes. Often a defect in one gene is not enough to elicit a full-blown disease, but when it occurs in the presence of other moderate defects, a disease becomes clinically manifest. It is the expectation of human geneticists that many chronic diseases can be explained by the combination of defects in a few (major) genes.
- Polygenic disease: This is third category of genetic disorder. According to the polygenic hypothesis, many mild defects in genes conspire to produce some chronic diseases. To date, the full genetic basis of polygenic diseases has not been worked out as multiple interacting defects are highly complex.

In case of association analysis, we need to have selection of representative samples from the population of interest and complete and accurate genotype data set. Therefore, in this statistical analysis representative sample can be selected using appropriate sampling procedure depending on cost of experiments. However, missing values in genotypes is non-avoidable.

Therefore, we need to employ appropriate imputation techniques. Brief descriptions about these two techniques are given in subsequent paragraphs.

Sampling Techniques

The genesis of multiphase design for case control studies are from sample surveys. Initially, two phase sampling was introduced by Neyman (1938) as a technique for stratification. In this technique researcher needs to draw a Simple Random Sample from the target population and classify objects into homogeneous strata. Further, subsamples from each stratum are drawn and observations on variable are recorded only on these sub-samples drawn in the second phase. With judicious choice of strata and optimum sampling ratios, these designs are very cost efficient. The basic idea of these designs is to use information available on all subjects in the main study and draw more informative sub-samples for additional, more expensive, measurement and combining the information from both phases in the analysis. This concept for in Genome Wide Association Studies (GWAS) was introduced by Satagopan et. al. (2002),. Previously, this design has been cited in epidemiologic literature by White (1982). The basic goal of two phase design is to maximize the power to detect gene and disease association when the main design constraint is the total cost. Mainly, this total cost depends on number of gene evaluations rather than total number of individuals. Therefore, in the first phase, all the genes of our interest are evaluated on a sub-set of individuals. Later, most promising genes are evaluated on additional/same subjects in the second phase. This will eliminate the wastage of resources on genes, which are not likely to be associated with a particular trait. In this situation we find two types of cases i.e. (i) when genes are co-related (ii) when genes are independent.

Let us assume the unit cost per gene evaluation and let T denotes the total number of genetic evaluation or total cost. Let, in a genome, there are m genetic loci. Consider very simple situation that out of these m gene only one gene is associated with the trait (disease) under consideration. Now our problem is to identify this true gene which is associated with our trait. Let there are N individuals which are available. In absence of any cost constraints the best way is to evaluate all m markers for all N subjects with a total cost of mN. The best way of testing association with the trait under this situation is making 2 X 2 table for each locus with presence or absence of trait as rows and alleles as columns then apply chi-square test for association. The target gene would be selected based on largest test statistics. Now let us assume a situation when $T < mN$, then it is not possible to evaluate all m markers but only T/m individuals can be evaluated at first stage, but selection of T/m individuals should be in such that the possibility of missing true gene associated with trait is minimum. Therefore, this design needs to be optimized for two stage selection.

In this design, all m genes are evaluated on n_1 individuals, where proportion of cases with trait and control remains the same as in the case of N individuals. After application of test statistics, rank them based on absolute value of test statistics. In the second stage, select top mi genes

where "i" is proportion of genes on sub-sequent subjects till cost is T through selection of same proportion of case and control subject as in the original population. Now, the problem boils down to determination of value n_1 and i (ith proportion) so that it leads to maximum probability of selecting true gene i.e. maximum power P of the statistics. Then $T = n_1m + n_2mi$ where, n_2 is numbers of subjects at the second stage. Now our aim is to maximize P with respect to ni and i subject to fixed T and m. Since, $T/ m= n_1 + n_{2i}$ = fixed, therefore, choosing n_1 and i determines n_2. In other word, optimization of power for two stage design can be seen equivalently, as determination of proportion of resources at the first stage i.e. $j= n_1m/T$ and determination of proportion "i" of the genes to be evaluated at the second stage. The proportion of total number of subject required for two stage design is given by j+(1-j)/i.

In other words, P can be written as P1 * P2 under the assumption that mutational profiles of all genes are mutually uncorrelated. Hence, P1 is the probability that true gene is among top i^{th} proportion in stage 1 and P2 is the probability that true gene has highest association among all null genes at stage-2. These probabilities can be calculated using statistical distributions (may be Gaussian approximation). In practice, the assumption of independent gene outcome may not be true within individual in case of testing multiple markers. These outcomes may be correlated due to various factors, such as genetic linkages and loss of heterozygosis, allele frequency, and marker density. The correlation due to recombination can be easily quantified and further these can be modelled through statistical distributions. Further these probabilities can be further evaluated using Mote Carlo simulation for different values of i, j and μ (mean). This design can be further extended for optimal design for more than one true gene.

Genotype Imputation

Identification and characterization of genetic variation of a species which affects its important traits are very important for increasing production and productivity in agriculture especially in the context of development of improved biotic and abiotic resistance breeds/varieties. The basic idea is that, data on a modest set of genetic variant measured in number of related subjects can provide useful information about other genetic variants in those subjects forms the theoretical under pinning of both genetic linkage mapping in pedigree and haplotype mapping in founder population. These studies typically used few markers to survey entire genome through identification of parts of chromosomes inherited from common ancestor. Earlier in genetic linkage and haplotype mapping, it was expected that long sketches of shared chromosome inherited from a relatively recent common ancestor. Sometimes, the focus of GWAS is on unrelated individuals and it expected to have small stretch on shared chromosome. Under these circumstances genotype imputation can use these short stretches of shared haplotype to estimate with great precision the effects of many variants that are not directly genotyped.

There are two broad categories of genotype imputation. First, imputing missing genotype from information on close relatives and second, genotype imputation from distant relatives. If it is

known that the haplotype individuals carried at every point on the genome and SNP alleles are also known within each unique population haplotype then it is possible to impute genotypes which a individual carries for any SNP locus. Genotype imputation is important due to following reasons:-

- In case of accurate SNP array technology also, large number of SNP genotype are missing which poses problems in genomic selection and GWAS.
- Genotype imputation can be used to get high density genotype when subject has been genotyped with low density array.
- It is quite useful for combining data sets genotyped from two different panels with sufficient overlap between panels.
- Genotype imputation is applied to recover genotype from full genome sequence data. (i.e. from very dense SNP/insertion and deletion, CNV for genomic predictions and GWAS).

There are number of approaches/tools for imputing missing Genotypes such as PHASE (IMPUTE 1.0, IMPUTE 2.0) FastPHASE, MACH, BEAGLE etc. But PHASE and Fast PHASE are most widely used. In case of genotype imputation number of tools uses Hidden Marker Model (HMM) at the backend. These basic approaches relies that if it is known that a particular SNP alleles are associated with a particular hyplotype in a population then it is possible to infer or impute genotype carried by the individual of same hyplotype for which it is not known. In case of HMM, the hidden state generates true hyplotypes in the population for which genotypes are known. Then HMM can be used to estimate the probability that an individual carries a particular genotype at a particular locus given the genotype data for that individual at other locus and rest of the population. Basically it takes advantage of a reference population which is densely genotyped at all SNP. The methods of imputation differ in their assumptions about the hidden states, the way state transition probabilities are derived, emission probability and the initial state probabilities.

Association Analysis

The association analysis can be taken up with well-defined phenotype of a population, and genotypes data set which is collected using sound techniques. The preliminary analysis of genome-wide association data is a series of single-locus statistic tests, examining each SNP independently for association to the phenotype. The statistical test conducted depends on a variety of factors, but first and foremost, statistical tests are different for quantitative traits versus case/control studies. Quantitative traits are generally analysed using generalized linear model (GLM) approaches and most commonly the Analysis of Variance (ANOVA), which is similar to linear regression with a categorical predictor variable of genotype classes. The null hypothesis of an ANOVA using a single SNP is that there is no difference between the trait

means of any genotype group. The assumptions of GLM and ANOVA are that (i) the trait is normally distributed, (ii) the trait variance within each group is the same, and (iii) the groups are independent. Dichotomous case/control traits are generally analysed using either contingency table methods or logistic regression. Contingency table tests examine and measure the deviation from independence that is expected under the null hypothesis that there is no association between the phenotype and genotype classes using Chi-square test and related Fisher's exact test. Logistic regression is an extension of linear regression where the outcome of a linear model is transformed using a logistic function that predicts the probability of having case status given a genotype class. Logistic regression is often the preferred approach because it allows for adjustment for clinical covariates (and other factors), and can provide adjusted odds ratios as a measure of effect size. Logistic regression has been extensively developed, and numerous diagnostic procedures are available to aid interpretation of the model. For both quantitative and dichotomous trait analysis (regardless of the analysis method), there are a variety of ways that genotype data can be encoded or shaped for association tests. The choice of data encoding can have implications for the statistical power of a test, as the degrees of freedom for the test may change depending on the number of genotype-based groups that are formed. Allelic association tests examine the association between one allele of the SNP and the phenotype. Genotypic association tests examine the association between genotypes (or genotype classes) and the phenotype. The genotypes for a SNP can also be grouped into genotype classes or models, such as dominant, recessive, multiplicative, or additive models. Each model makes different assumptions about the genetic effect in the data by assuming two alleles for a SNP, A and a, a dominant model (for A) assumes that having one or more copies of the A allele increases risk compared to a (i.e. Aa or AA genotypes have higher risk). The recessive model (for A) assumes that two copies of the A allele are required to alter risk, so individuals with the AA genotype are compared to individuals with Aa and aa genotypes. The multiplicative model (for A) assumes that if there is 3 x risk for having a single A allele, there is a 9 x risk for having two copies of the A allele. In this case, if the risk for Aa is k, the risk for AA is k^2. The additive model (for A) assumes that there is a uniform, linear increase in risk for each copy of the A allele, so if the risk is 3 x for Aa, there is a 6x risk for AA. In this case, the risk for Aa is k and the risk for AA is 2k. A common practice for GWAS is to examine additive models only, as the additive model has reasonable power to detect both additive and dominant effects, but it is important to note that an additive model may be underpowered to detect some recessive effects. Rather than choosing one model a priori, some studies evaluate multiple genetic models coupled with an appropriate correction for multiple testing.

Acknowledgements

All contents of this lecture notes are taken from different web resources including research articles, presentations, lecture notes etc. We duly acknowledge contributions of all these resources.

Genomic Selection

(Practical)

Genomic selection is emerging as an efficient and cost-effective method for estimating breeding values using molecular markers distributed over the entire genome. Basically, it involves estimating the simultaneous effects of all genes and combining the estimates to predict the total genomic estimated breeding value (GEBV).

Let's consider a column vector y containing the phenotypic values for a trait measured in n individuals. It is assumed that these observations are described adequately by a linear model with a p×1 vector of fixed effects (β) of environments/locations and a q×1 vector of random effects (u) of SNPs. In matrix form,

where Var(u) = $K\sigma^2_u$ and the residual variance is Var(e) = $I\sigma^2_e$. This class of mixed models, in

$$\boldsymbol{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{u} + \boldsymbol{e}$$

which there is a single variance component other than the residual error, has a close relationship with ridge regression (ridge parameter $\lambda = \sigma^2_e / \sigma^2_u$).

Genomic Selection using `BGLR` (Method: `BayesA`)

```
library(BGLR)
setwd("")
x<-read.table(file="genotype.txt")
dim(x)
y<-as.matrix(read.table(file="phenotype.txt"))
dim(y)
n<-nrow(x)
p<-ncol(x)
X<-x[1:n,1:p]
for(i in 1:p)
{
        X[,i]<-(X[,i]-mean(X[,i]))/sd(X[,i])
}
nIter=50000;
burnIn=5000;
thin=10;
R2=0.5;
S0=NULL;
weights=NULL;
```

```
ETA<-list(list(X=X,model='BayesA'))
fit_BA=BGLR(y=y,ETA=ETA,nIter=nIter,burnIn=burnIn,thin=thin,df0=5,S0=S0,weights=w
eights,R2=R2)
fit_BA$yHat
cor(fit_BA$yHat,y)
```

Genomic Selection using `BGLR` (Method: `BayesB`)

```
install.packages("BGLR")
library(BGLR)
setwd("")
x<-read.table(file="genotype.txt")
dim(x)
y<-as.matrix(read.table(file="phenotype.txt"))
dim(y)
n<-nrow(x)
p<-ncol(x)
X<-x[1:n,1:p]
for(i in 1:p)
{
        X[,i]<-(X[,i]-mean(X[,i]))/sd(X[,i])
}
nIter=50000;
burnIn=5000;
thin=10;
R2=0.5;
S0=NULL;
weights=NULL;
ETA<-list(list(X=X,model='BayesB',probIn=0.05))
fit_BB=BGLR(y=y,ETA=ETA,nIter=nIter,burnIn=burnIn,thin=thin,df0=5,S0=S0,weights=w
eights,R2=R2)
fit_BB$yHat
cor(fit_BB$yHat,y)
```

Genomic Selection using `BGLR` (Method: `BayesB`)

```
library(BGLR)
setwd("")
x<-as.matrix(read.table(file="genotype.txt"))
dim(x)
y<-read.table(file="phenotype.txt")
head(y)
```

```
dim(y)
y<-y[,1]
n<-nrow(x)
p<-ncol(x)
yNA<-y
x<-scale(x,center=TRUE,scale=TRUE)
x[1:10,1:10]
G<-tcrossprod(x)/p
G[1:10,1:10]
ETA<-list(list(K=G,model='RKHS'))
fm<-BGLR(y=yNA,ETA=ETA,nIter=50000,
burnIn=5000,thin=10,S0=NULL,weights=NULL,R2=0.5)
cor(fm$yHat,y)
fm$yHat
```

Genomic Selection by LASSO

```
library(glmnet)
library(Matrix)
library(foreach)
setwd("")
x<-as.matrix(read.table(file="genotype.txt"))
dim(x)
y<-read.table(file="phenotype.txt")
y=y[,1]
head(y)
fit<-glmnet(x,y,family="gaussian",alpha=1,nlambda=100)
print(fit)
cvfit=cv.glmnet(x,y,family="gaussian",type.measure="mse",nfolds=10)
print(cvfit)
z<-predict(cvfit,newx=x,s=c(cvfit$lambda.min))
cor(y,z)
z=z[,1]
```

Genomic Selection by Random Forest

```
library(randomForest)
setwd("")
x<-as.matrix(read.table(file="genotype.txt"))
dim(x)
y<-as.matrix(read.table(file="phenotype.txt"))
dim(y)
```

```
myrf<-randomForest(x,y,mtry=1807,ntree=1000,type="mse",replace=TRUE,nodesize=50)
myrf
z<-predict(myrf,x)
z
cor(z,y)
```

Genomic Selection by Ridge Regression

```
library(glmnet)
library(Matrix)
setwd("")
x<-as.matrix(read.table(file="genotype.txt"))
dim(x)
y<-read.table(file="phenotype.txt")
y=y[,1]
head(y)
fit<-glmnet(x,y,family="gaussian",alpha=0,nlambda=100)
print(fit)
cvfit=cv.glmnet(x,y,family="gaussian",type.measure="mse",nfolds=10)
print(cvfit)
z<-predict(cvfit,newx=x,s=c(cvfit$lambda.min))
cor(y,z)
z=z[,1]
```

Steps involving in GWAS and genomic selection using `rr-BLUP`

- Convert the genotype marker data into matrix form, where, the values are coded as -1,0, 1 format.

GWAS

```
> library(rrBLUP)
> #random population of 200 lines with 1000 markers
> M <- matrix(rep(0,200*1000),1000,200)
> for (i in 1:200) {
+   M[,i] <- ifelse(runif(1000)<0.5,-1,1)
+ }
```

```
> colnames(M) <- 1:200
> geno <- data.frame(marker=1:1000,chrom=rep(1,1000),pos=1:1000,M,check.names=FALSE)
>
> QTL <- 100*(1:5) #pick 5 QTL
> u <- rep(0,1000) #marker effects
> u[QTL] <- 1
> g <- as.vector(crossprod(M,u))
> h2 <- 0.5
> y <- g + rnorm(200,mean=0,sd=sqrt((1-h2)/h2*var(g)))
> pheno <- data.frame(line=1:200,y=y)
> scores <- GWAS(pheno,geno,plot=FALSE)
[1] "GWAS for trait: y"
[1] "Variance components estimated. Testing markers."

> head(scores)
  marker chrom pos         y
1      1     1   1 0.1025945
2      2     1   2 0.3458941
3      3     1   3 0.1682724
4      4     1   4 0.3916777
5      5     1   5 0.4491785
6      6     1   6 0.5247056
```

The GWAS function mainly takes two arguments pheno and geno.

- pheno is a data frame where the first column is the line name. The remaining columns can be either a phenotype or the levels of a fixed effect. Any column not designated as a fixed effect is assumed to be a phenotype.
- geno is data frame with the marker names in the first column.
- The GWAS function returns a data frame where the first three columns are the marker name, chromosome, and position, and subsequent columns are the marker scores for the trait.
- For the prediction of the marker effects, mixed.solve is used with the following important arguments.

y	Vector (n × 1) of observations. Missing values (NA) are omitted, along with the corresponding rows of X and Z.
Z	Design matrix (n × m) for the random effects. If not passed, assumed to be the identity matrix.
K	Covariance matrix (m × m) for random effects; must be positive semi-definite. If not passed, assumed to be the identity matrix.
X	Design matrix (n × p) for the fixed effects. If not passed, a vector of 1's is used to model the intercept. X must be full column rank (implies β is estimable).
method	Specifies whether the full ("ML") or restricted ("REML") maximum-likelihood method is used.

- mixed.solve function returns a bunch of values including estimator of σ^2_u by $Vu, estimator of σ^2_e by $Ve, Best Linear Unbiased Estimation (BLUE) of β by $beta, Best Linear Unbiased prediction (BLUP) of u by $u etc.
- BLUP of u gives the effects of all SNPs.
- By putting Z = M, and default value of K, marker effects can be obtained. Again by putting K = M and default value of Z, breeding values can be obtained

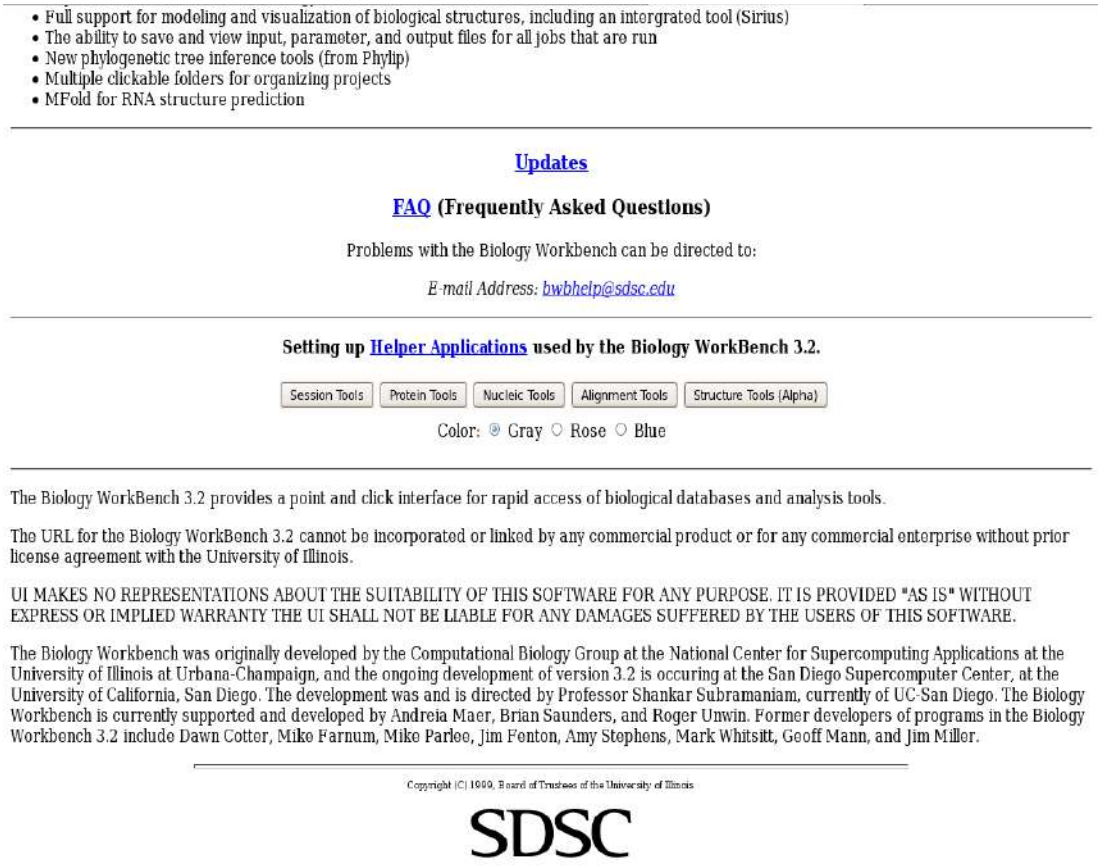
```
> M <- matrix(rep(0,200*1000),200,1000)
> for (i in 1:200) {
+   M[i,] <- ifelse(runif(1000)<0.5,-1,1)
+ }
> #random phenotypes
> u <- rnorm(1000)
> g <- as.vector(crossprod(t(M),u))
> h2 <- 0.5 #heritability
> y <- g + rnorm(200,mean=0,sd=sqrt((1-h2)/h2*var(g)))
> ans <- mixed.solve(y,Z=M) #By default K = I
> accuracy <- cor(u,ans$u)
> ans$Vu
[1] 0.3513168
> ans$Ve
[1] 1866.874
> ans$beta
[1] 1.771402
> dim(ans$u)
[1] 1000

> accuracy
[1] 0.3241786

> ans <- mixed.solve(y,K=A.mat(M))
> accuracy <- cor(g,ans$u)
> ans$Vu
[1] 174.8232
> ans$Ve
[1] 1866.874
> ans$beta
[1] 1.926871
> dim(ans$u)
[1] 200
```

Protein Sequences to Structure

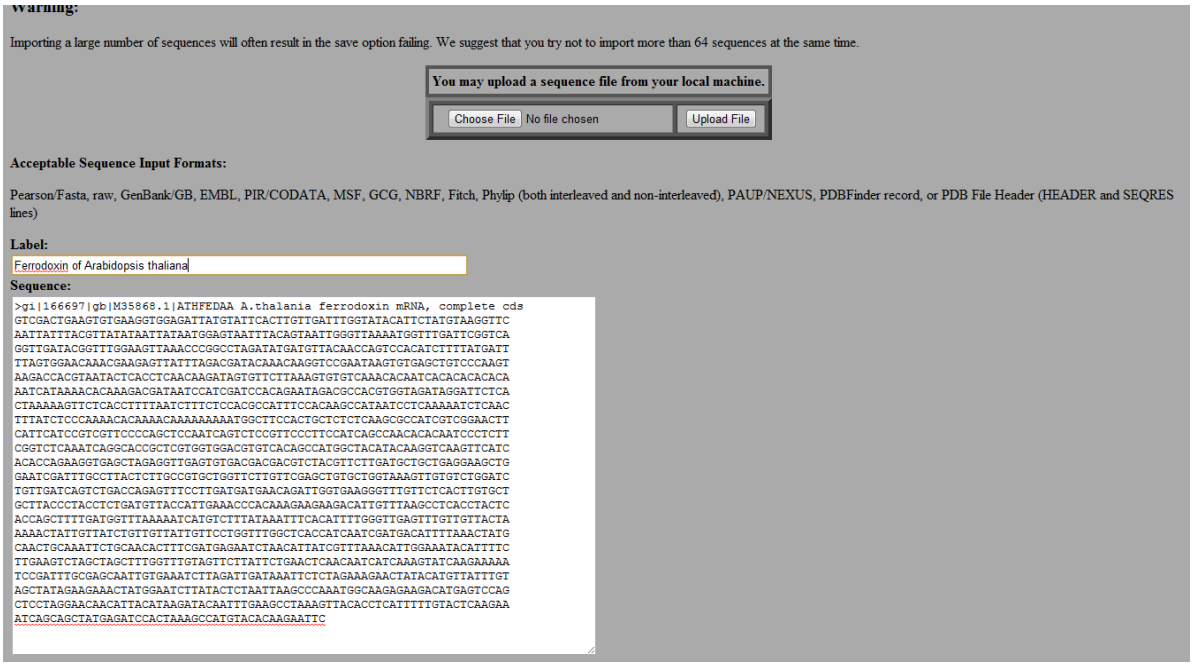
(Practical)

Step-1: Primary structure prediction

1. Log onto Biology Workbench by using the URL **http://workbench.sdsc.edu**. Though this is an open source tool, users need to register for a free account. For registration click on register link on the home page.



2. If you are a registered user, please click on “Click to Enter the Biology Workbench 3.2”.
3. Click on “Nucleic Tools”.
4. In the resulted page under the default session of nucleic tools section select “Add New Nucleic Sequence”.
5. Click on “Run”.

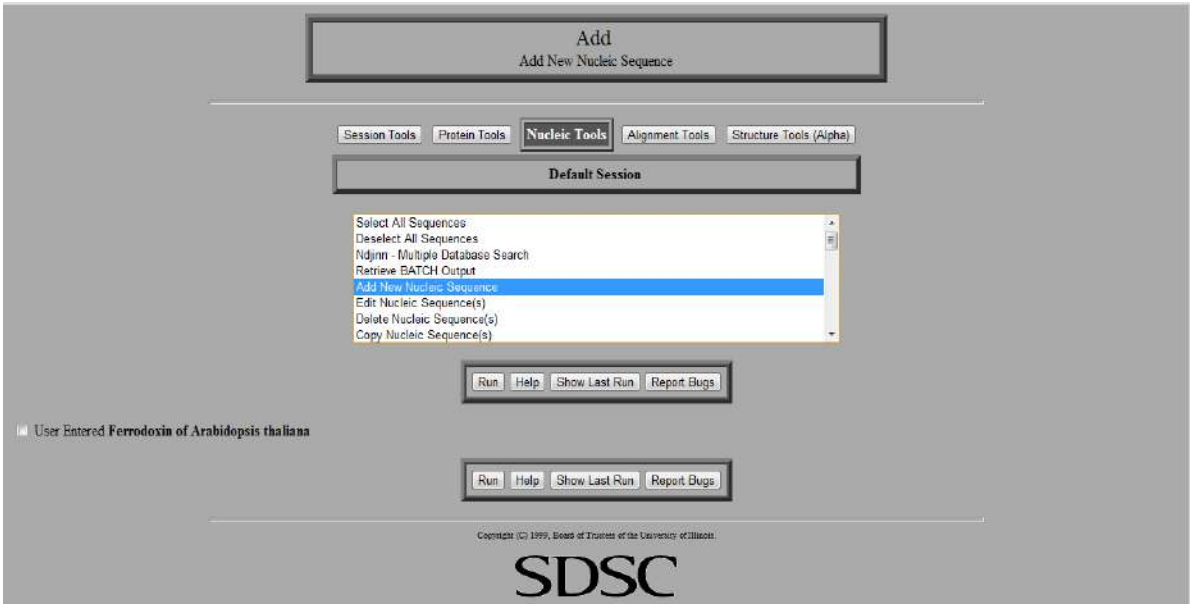


6. Download the mRNA sequence of Ferrodoxin enzyme of *Arabidopsis thaliana* from NCBI using the accession number M35868.1 in FASTA format as shown below in a separate window.

```
>gi|166697|gb| |ATHFEDAA A.thalania ferrodoxin mRNA, complete cds
GTCGACTGAAGTGTGAAGGTGGAGATTATGTATTCACTTGTTGATTTGGTATACATTCTATGTAAGGTTC
AATTATTTACGTTATATAATTATAATGGAGTAATTTACAGTAATTGGGTTAAAATGGTTTGATTCGGTCA
GGTTGATACGGTTTGGAAGTTAAACCCGGCCTAGATATGATGTTACAACCAGTCCACATCTTTTATGATT
TTAGTGGAACAAACGAAGAGTTATTTAGACGATACAAACAAGGTCCGAATAAGTGTGAGCTGTCCCAAGT
AAGACCACGTAATACTCACCTCAACAAGATAGTGTTCTTAAAGTGTGTCAAACACAATCACACACACACA
AATCATAAAACACAAAGACGATAATCCATCGATCCACAGAATAGACGCCACGTGGTAGATAGGATTCTCA
CTAAAAAGTTCTCACCTTTTAATCTTTCTCCACGCCATTTCCACAAGCCATAATCCTCAAAAATCTCAAC
TTTATCTCCCAAAACACAAAACAAAAAAAAATGGCTTCCACTGCTCTCTCAAGCGCCATCGTCGGAACTT
CATTCATCCGTCGTTCCCCAGCTCCAATCAGTCTCCGTTCCCTTCCATCAGCCAACACACAATCCCTCTT
CGGTCTCAAATCAGGCACCGCTCGTGGTGGACGTGTCACAGCCATGGCTACATACAAGGTCAAGTTCATC
ACACCAGAAGGTGAGCTAGAGGTTGAGTGTGACGACGACGTCTACGTTCTTGATGCTGCTGAGGAAGCTG
GAATCGATTTGCCTTACTCTTGCCGTGCTGGTTCTTGTTCGAGCTGTGCTGGTAAAGTTGTGTCTGGATC
TGTTGATCAGTCTGACCAGAGTTTCCTTGATGATGAACAGATTGGTGAAGGGTTTGTTCTCACTTGTGCT
GCTTACCCTACCTCTGATGTTACCATTGAAACCCACAAAGAAGAAGACATTGTTTAAGCCTCACCTACTC
ACCAGCTTTTGATGGTTTAAAAATCATGTCTTTATAAATTTCACATTTTGGGTTGAGTTTGTTGTTACTA
AAAACTATTGTTATCTGTTGTTATTGTTCCTGGTTTGGCTCACCATCAATCGATGACATTTTAAACTATG
CAACTGCAAATTCTGCAACACTTTCGATGAGAATCTAACATTATCGTTTAAACATTGGAAATACATTTTC
TTGAAGTCTAGCTAGCTTTGGTTTGTAGTTCTTATTCTGAACTCAACAATCATCAAAGTATCAAGAAAAA
TCCGATTTGCGAGCAATTGTGAAATCTTAGATTGATAAATTCTCTAGAAAGAACTATACATGTTATTTGT
AGCTATAGAAGAAACTATGGAATCTTATACTCTAATTAAGCCCAAATGGCAAGAGAAGACATGAGTCCAG
CTCCTAGGAACAACATTACATAAGATACAATTTGAAGCCTAAAGTTACACCTCATTTTTGTACTCAAGAA
ATCAGCAGCTATGAGATCCACTAAAGCCATGTACACAAGAATTC
```

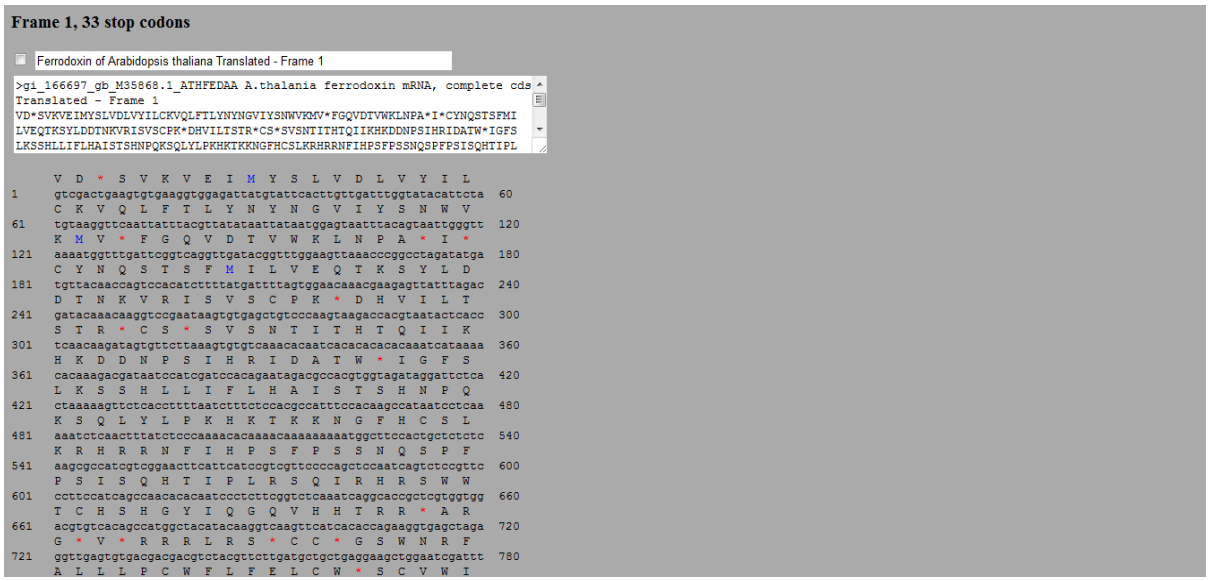
7. Copy the sequence in fasta format.
8. In the Add New Nucleic Sequence page of Biology Workbench paste the sequence in "Sequence" text area and type "Ferrodoxin of *Arabidopsis thaliana*" in the "Label" text box or the fasta file downloaded from NCBI can also be uploaded by using its browse option.
9. Click on save. The sequence label will come as list in the same page with a check box.
10. Click on the check box to select the sequence.

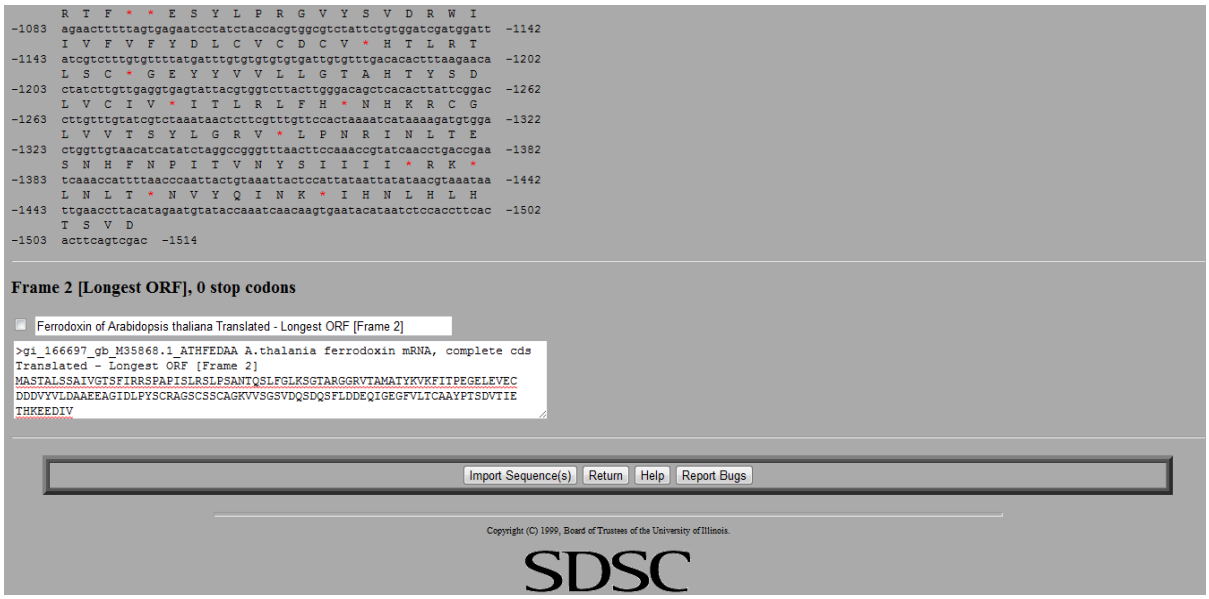
11. Select the option "SIXFRAME-Generate & Import 6 Frame Translations on a NS" from the "Default Session" and click on the "Run" button.



12. The resulted page provides a form for filling the parameters of SIXFRAME, i.e. translation table, sequence positions, Frames to translate etc. as shown below.

13. Allowing the default values click on Submit button.
14. The final result page gives all the six frame translations of the input mRNA sequence and information about the longest ORF is given at the bottom of the window





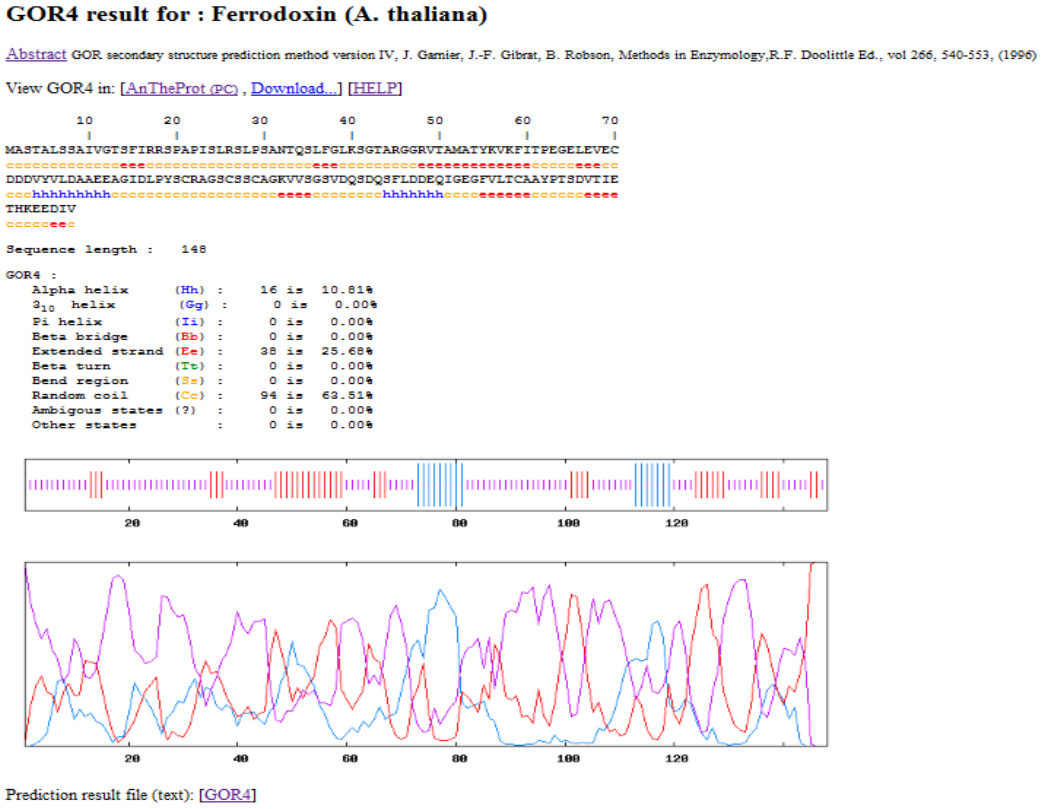
15. Save the protein sequence from the longest ORF in a separate text file as *fer.txt* for further analysis.

Step-2: Secondary structure prediction

1. To predict the secondary structure by GOR IV tool open the URL http://npsa-pbil.ibcp.fr/cgi-bin/npsa_automat.pl?page=npsa_gor4.html
2. Copy the sequence from *fer.txt.*
3. Paste the sequence in the input window and name the sequence as "Ferrodoxin (*A. thaliana)"*

4. Click on Submit.

5. This result page shows the output as follows giving the information about the helices, sheets and coils.



Step-3: Tertiary structure prediction

(a) By *ab-initio* method

By QUARK ONLINE

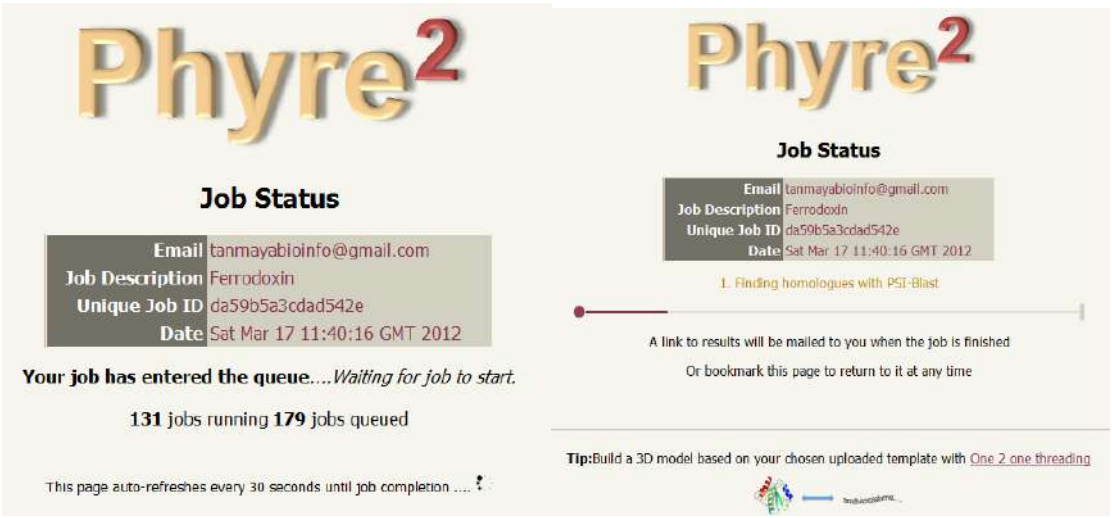
1. To open the "QUARK ONLINE" open the URL **http://zhanglab.ccmb.med.umich.edu/QUARK/**.
2. Paste the protein sequence of predicted protein from the file *fer.txt* sequence labeling the sequence as its name.
3. Provide your email address in the appropriate box to which the result will be mailed.

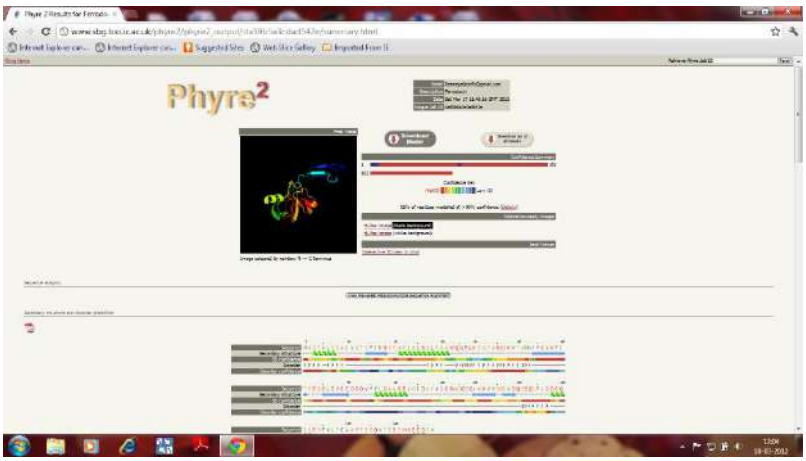
4. Click on "Run QUARK".
5. The results will be mailed to the email address provided after the completion of the job and can also be downloaded from the result page.
6. Check your email ID to get the link of the PDB file.

(b) By fold recognition method

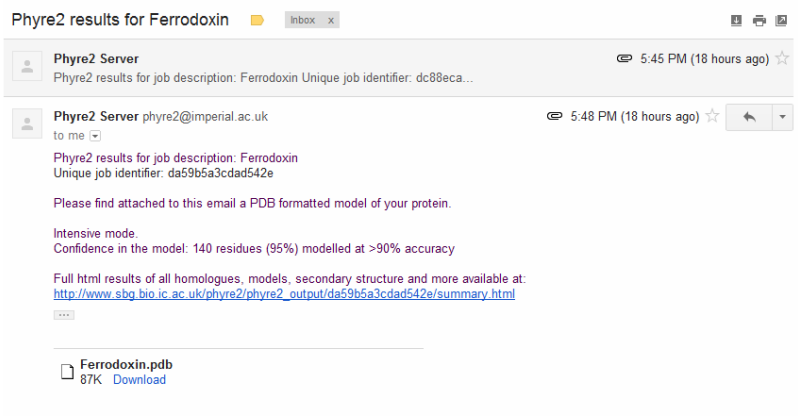
1. Open PHYRE from the URL http://www.sbg.bio.ic.ac.uk/phyre2/html/page.cgi?id=index
2. Paste the predicted sequence of Ferrodoxin of *A. thaliana* from *fer.txt* and label it as its name.
3. Provide your email ID for the result.

4. Now click on "Phyre Search". The result page will be displayed as follows showing the predicted secondary structure and the 10 best tertiary protein models based on fold recognition.





5. Check your email ID to get the coordinates of the best modeled structure, paste the same in a notepad and save the file in .pdb extension



6. The saved PDB file can be visualized using Accelrys Discovery Studio 2.0. Further refinement and validation can be done by ModLoop and Rampage servers respectively. Both are discussed in "Step – 4".

(c) By Homology Modeling using MODELLER.

Modeller requires three different input files which are to be prepared WITH OUT ANY ERROR as mentioned below.

1. PDB file of known structure (*.pdb)
2. Alignment file of target and template (*.ali)
3. Python Script file for generating model (*.py)

Softwares Required: Modeller 9v8, Accelrys DS Visualizer 2.0, ClustalX, Python 2.6.5

Preparing alignment and PDB file

1. Open *fer.txt* change it in fasta format by adding ">Ferrodoxin" above the sequence and copy all.
2. Open the NCBI BLAST (http://blast.ncbi.nlm.nih.gov/Blast.cgi) and click on protein blast and paste the copied sequence in the sequence text area.
3. On the BLAST page come to the option for changing database and set the database as "Protein Data Bank (PDB)".
4. Allowing the other default values click on BLAST.
5. On the result page find out the most similar sequences by highest bit score and lowest e-value having a similarity of more than 35% with the query.
6. In this case the template selected is 1PFD- Chain A, The Solution Structure Of High Plant Parsley [2fe-2s] Ferredoxin, Nmr, 18 Structures.
7. Download the PDB structure of template i.e. 1PFD.pdb from http://www.pdb.org and put the file in the same folder where *fer.txt* is present.
8. Open the structure in Accelrys DS Visualizer 2.0 and delete all the chains except Chain A and then click on Sequence ->Show sequence
9. Copy the sequence and paste it in *fer.txt* as follows and save.

```
>Ferrodoxin
MASTALSSAIVGTSFIRRSPAPISLRSLPSANTQSLFGLKSGTARGGRVTAMATYKVKFITPEGEL
EVEC
DDDVYVLDAAEEAGIDLPYSCRAGSCSSCAGKVVSGSVDQSDQSFLDDEQIGEGFVLTCAAYPTSD
VTIE
THKEEDIV
>1PFD
ESVHDFTVKDAKENDVDLSIFKGKVLLIVNVASKCGMTNSNYAEMNQLYEKYKDQGLEILAFPCNQ
FGEEEPGTNDQITDFVCTRFKSEFPIFDKIDVNGENASPLYRFLKLGKWGIFGDDIQWNFAKFLVN
KDGQVVDRY
```

fer.txt file

10. Open ClustalX2 from start menu, choose Load Sequences of File menu and browse *fer.txt*.
11. After uploading click "Alignment" then click "Output Format Options" where a sub-window displayed on your monitor to adjust the parameters.
    a. Un-tick CLUSTAL format.
    b. Tick on NBRF/PIR format.
    c. Select Parameters output as "On" from the dropdown.
    d. Click on "OK".

12. Click on “Do Complete Alignment” sub-menu from “Alignment” menu.
13. Now your output files will be created in your folder (You can also change the location of output file from dialogue box appeared by clicking on “Do Complete Alignment”).
14. Open the output file, having .pir extension in word pad and save the file in .ali extension (e.g. f*er.ali*).
15. Then do modifications in *fer.ali* file by observing standard file, which is given below.

```
>P1;Ferrodoxin
sequence:Ferrodoxin::::::::
MASTALSSAIVGTSFIRRSPAPISLRSLPSANTQSLFGLKSGTARGGRVTAMATYKVKFI
TPEGELEVECDDDVYVLDAAEEAGIDLPYSCRAGSCSSCAGKVVSGSVDQSDQSFLDDEQ
IGEGFVLTCAAYPTSDVTIETHKEEDIV
*
>P1;1PFD
structureM:1PFD:1:A:96:A:THE SOLUTION STRUCTURE OF HIGH PLANT
PARSLEY [2FE-2S] FERREDOXIN, NMR, 18 STRUCTURES:Petroselinum
crispum ::
--------------------------------------------------ATYNVKLI
TPDGEVEFKCDDDVYVLDQAEEEGIDIPYSCRAGSCSSCAGKVVSGSIDQSDQSFLDDEQ
MDAGYVLTCHAYPTSDVVIETHKEEEIV
*
```

fer.ali file

2. Preparing Script files

1. Prepare the script file, as follows and save it as mod.py in the same folder

```
# Homology modeling by the automodel class
from modeller import *              # Load standard Modeller classes
from modeller.automodel import *    # Load the automodel class

log.verbose()    # request verbose output
env = environ()  # create a new MODELLER environment to build this
model in

# directories for input atom files
env.io.atom_files_directory = ['.', '1PFD.pdb']

a = automodel(env,
              alnfile  = 'fer.ali',     # alignment filename
              knowns   = '1PFD',              # codes of the
templates
              sequence = 'Ferrodoxin')              # code of the
target
a.starting_model= 1                 # index of the first model
a.ending_model  =5 # index of the last model
                                    # (determines how many models to
calculate)
a.make()                            # do the actual homology
modeling
```

mod.py File (script file)

3. Running Modeller

1. Double click on the mod.py file (Installation of python is necessary)

   OR

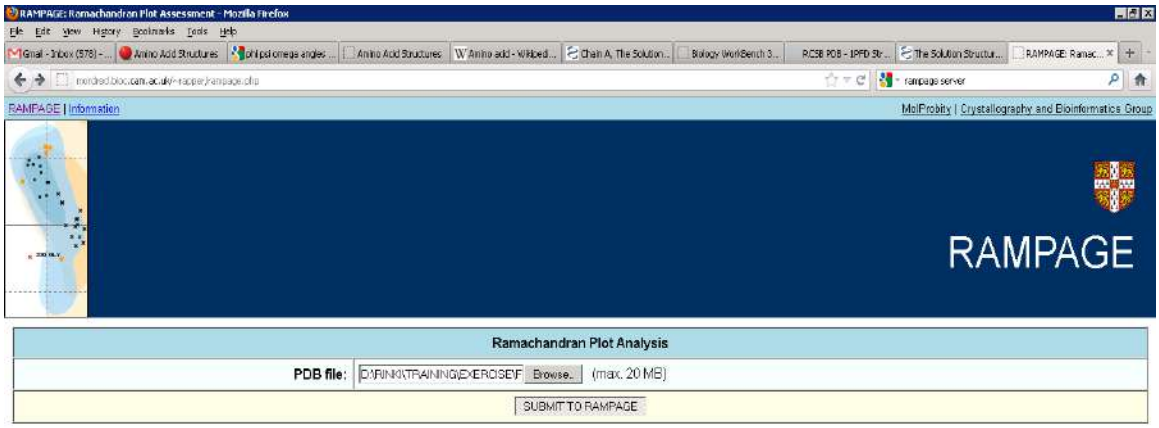
   If python is not installed please follow the steps below

   a) Copy all three (*.ali, *.py & *.pdb) and paste it in C:\Program Files\Modeller9v10\bin.
   b) Then open Start->All Programs->Modeller9v10->Modeller
   c) in the command prompt type cd bin then press Enter and again type the command > **mod9v10 mod.py**
2. Out put file will be created automatically in the same directory within approximately 3-5 min of analysis based on the length of the query sequence.
3. Copy the output file having the extension .B99990001.pdb and paste it in your folder then save the same file as Ferrodoxin.pdb
4. Visualize your 3D model created in Accelrys DS Visualizer 2.0.

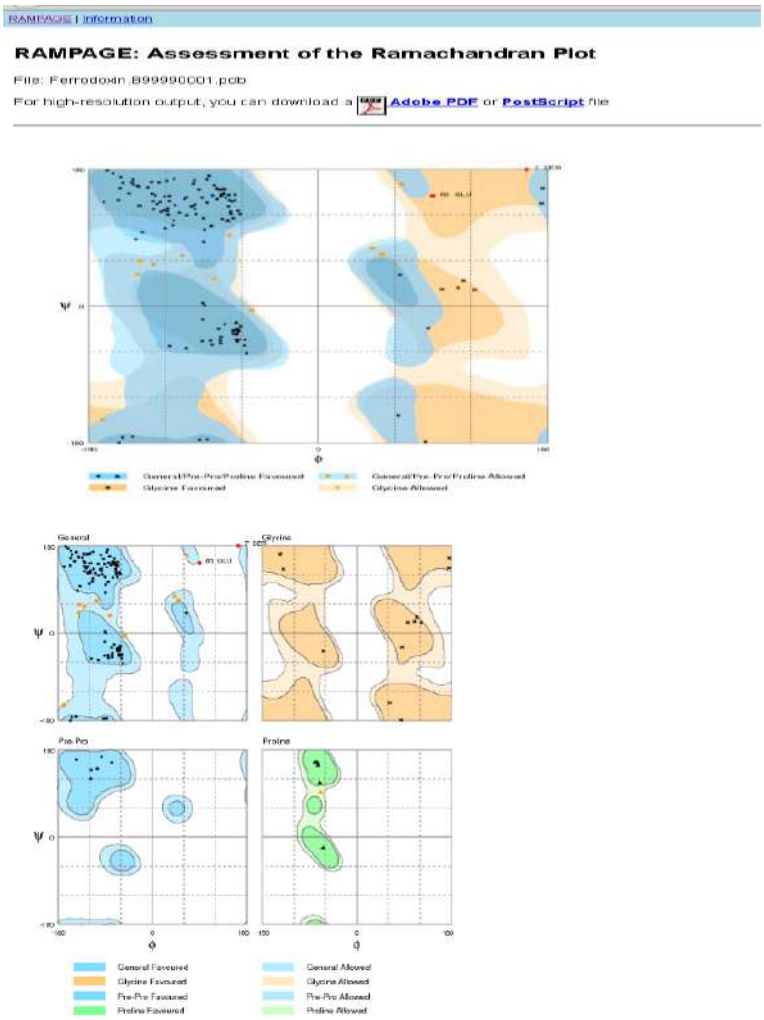
Step-4: Structure validation and Loop refinement

1. Open the URL http://mordred.bioc.cam.ac.uk/~rapper/rampage.php for RAMPAGE server

Note: This server validates the protein structure by Ramchandran plot and the plots for glycine and proline.



2. Browse the PDB file in the file browser option and click on “SUBMIT TO RAMPAGE” button.



RAMPAGE | Information

RAMPAGE: Assessment of the Ramachandran Plot

File: Ferrodoxin.B99990001.pdb

For high-resolution output, you can download a Adobe PDF or PostScript file

Evaluation of residues

```
Residue [   18 :ARG] (  65.53, 150.37| in Allowed region
Residue [   37 :PHE] (-105.52,  66.55| in Allowed region
Residue [   54 :THR] (  42.52,  76.42| in Allowed region
Residue [   76 :VAL] ( -81.93,  36.47| in Allowed region
Residue [   88 :PRO] ( -69.73,  93.63| in Allowed region
Residue [   92 :ARG] (-129.37,  54.69| in Allowed region
Residue [   97 :SER] (-140.12,  59.17| in Allowed region
Residue [   98 :SER] (-169.26,-149.51| in Allowed region
Residue [   99 :CYS] (  59.15,  58.30| in Allowed region
Residue [  111 :SER] ( -52.32,  -4.99| in Allowed region
Residue [  114 :SER] (-141.57,  41.61| in Allowed region
Residue [    7 :SER] ( 163.53, 179.45| in Outlier region
Residue [   55 :GLU] (  89.57, 194.67| in Outlier region
Number of residues in favoured region    (~98.0% expected)  :  133 ( 91.1%)
Number of residues in allowed region     ( ~2.0% expected)  :   11 (  7.5%)
Number of residues in outlier region                        :    2 (  1.4%)
```

RAMPAGE by Paul de Bakker and Simon Lovell.

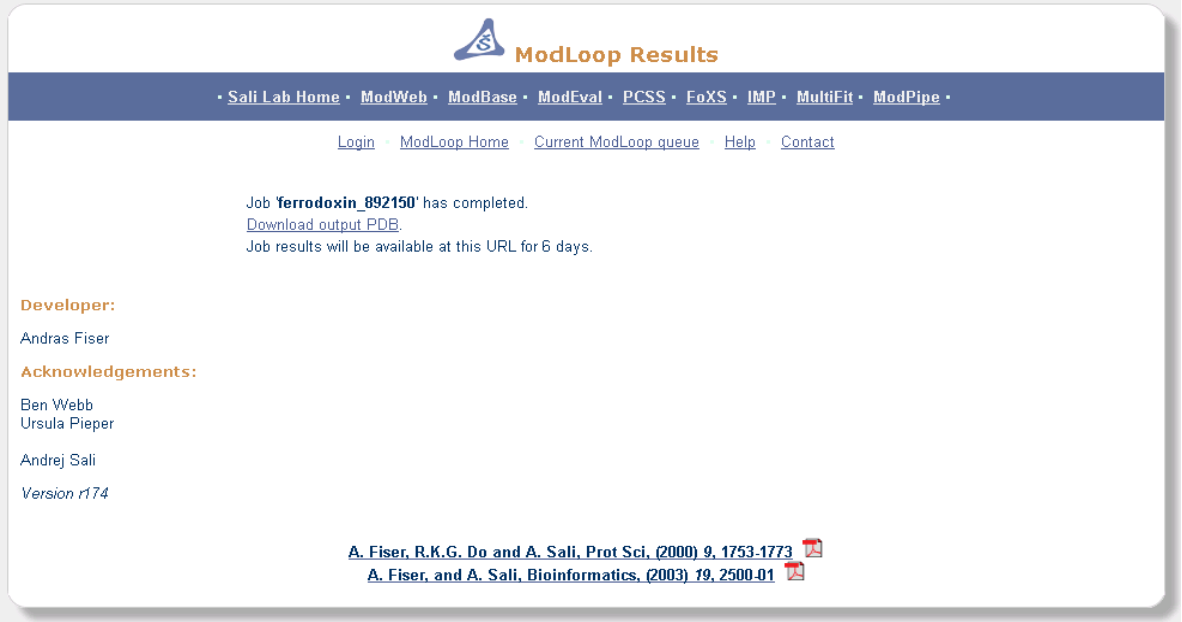
Please cite: S.C. Lovell, I.W. Davis, W.B. Arendall III, P.I.W. de Bakker, J.M. Word, M.G. Prisant, J.S. Genetics **50**: 437-450.

3. Check the result page for quality of the predicted structure under the “Evaluation of residues” heading. For individual residues plots can be checked.

4. If you are finding any residue present in the disallowed region then go for loop refinement using ModLoop server of Sali Lab (http://modbase.compbio.ucsf.edu/modloop/).

5. Fill your email id in the first text box (this is optional)
6. Fill “MODELIRANJE” in the license key text box
7. Browse your PDB file in the file browser
8. Give the range of 7-10 residues between which the residue number in disallowed region falls (eg. for 7give the range like 5::10::) in the text area for loop segments.
9. Give any name for your model in the name your model text box and click on process.

Note: It will be redirected to a page which will give you a JOB ID and a link to the result page. By clicking on that, it will again redirect to another page which will show the job status. After completion of your job at ModLoop server, a link to download the resulted **PDB file** will be provided on the same page.

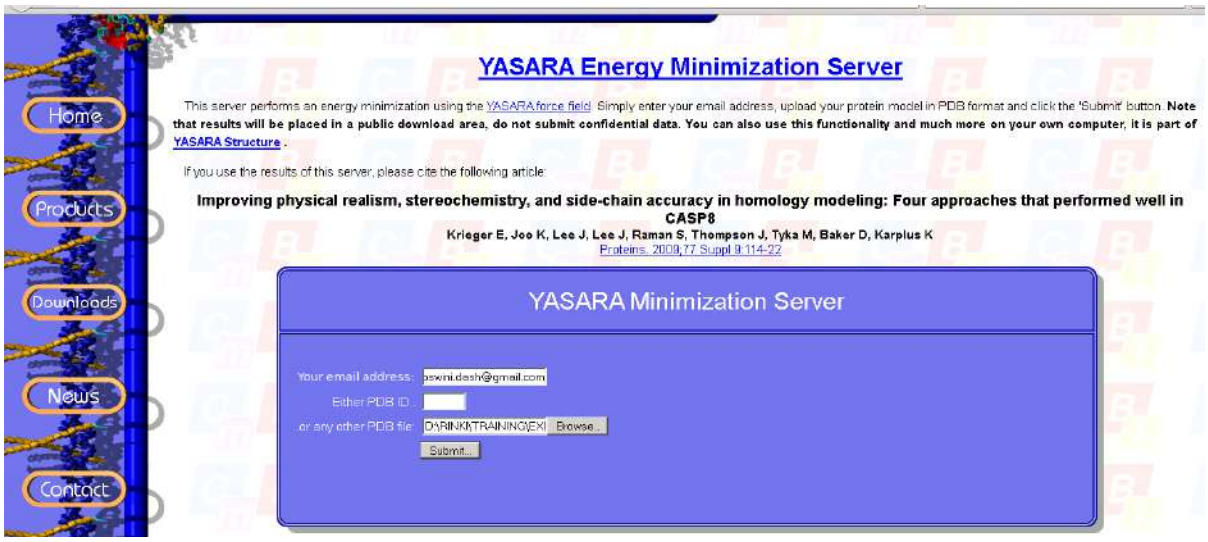


10. Download the PDB file, save it and check it in Rampage server for the structure validation.
11. All the models generated by *ab-intio* and fold recognition may also be refined and verified for comparison.
12. Rename the model as "structure" after loop refinement.

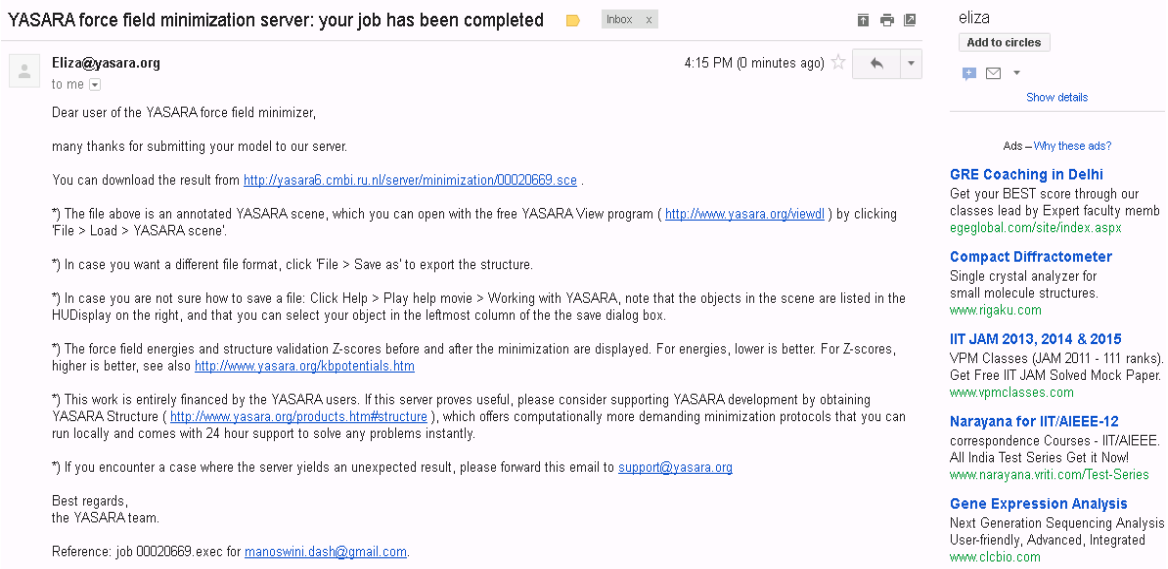
    Note:- The above steps of loop refinement need to be repeated until 98% residues lied in the allowed region.

Step-5: Energy minimization of loop refined structure

1. Open URL http://www.yasara.org/minimizationserver.htm
2. Give your mail ID and browse the PDB file which has already been loop refined then click "SUBMIT" button.

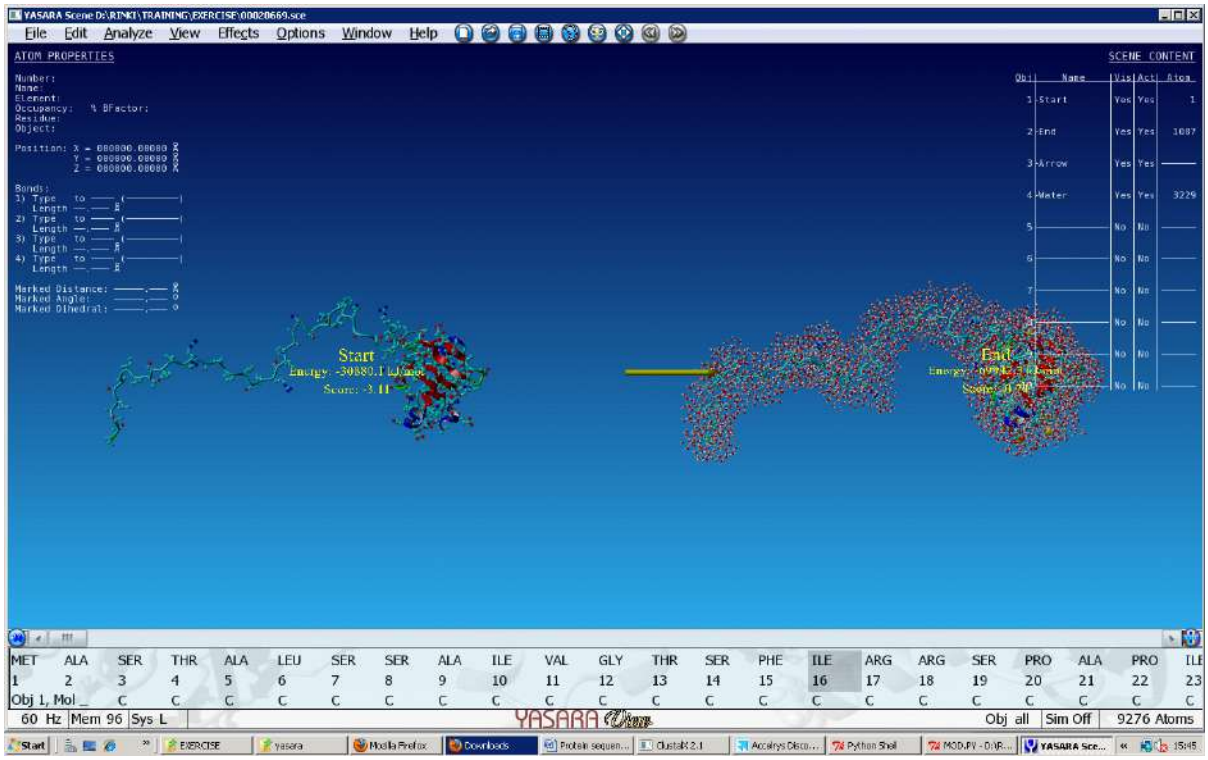


3. A confirmation mail followed by a job completion mail will be sent to the given email-ID.
4. Click on the link from the mail mentioning job has been completed and save it.



5. Open C:\yasara\yasara .
6. Go to file and then load followed YASARA scene. It will open a window where sce file will be browsed.

7. It will show energy and score of starting and end model.



8. For further work file will be saved in pdb format.

References

1. http://workbench.sdsc.edu
2. http://npsa-pbil.ibcp.fr/cgi-in/npsa_automat.pl?page=npsa_gor4.html
3. http://zhanglab.ccmb.med. umich.edu/QUARK
4. http://www.sbg.bio.ic.ac.uk/~phyre
5. http://blast.ncbi.nlm.nih.gov
6. http://www.pdb.org
7. http://mordred.bioc.cam.ac.uk/~rapper/rampage.php
8. http://modbase.compbio.ucsf.edu/modloop/
9. A.Fiser, R.K.G. Do and A. Sali, Prot Sci, (2000) *9*, 1753-1773
10. A.Fiser, and A. Sali, Bioinformatics, (2003) *18*, 2500-01
11. E. Krieger et.al. (2009) Improving physical realism, stereochemistry, and side-chain accuracy in homology modeling: Four approaches that performed well in CASP8 *Proteins*, *77*,114-122

Variant Analysis from RNA-Seq Data (Practical)

(1) Data Generation /Collection

• ***Step 1: Downloading SRA files from NCBI through command line***

wget -c -r ftp://ftp.ncbi.nlm.nih.gov/sra/sra-instant/reads/ByRun/sra/SRRfile -O /Anoop/SRA/file.sra

• ***Step 2: Conversion of .sra files to fastq***

/sratoolkit.2.8.2-1-centos_linux64/bin/fastq-dump -A SRRfile.sra -O SRR.fastq

(2) Data Pre-processing

Step 1: Trim adapters using trimomatic tool
java -jar /opt/software/Trimmomatic-0.32/trimmomatic-0.32.jar SE -phred33 -trimlog logfile /Anoop/fastq/SRR1772681.fastq SRR1772681_trimmed.fastq ILLUMINACLIP:/opt/software/Trimmomatic0.32/adapters/TruSeq2-SE.fa:2:30:10 LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15 MINLEN:36

Step 2: Build index of reference genome
Bowtie2 builds an index from the reference genome and then aligns the reads against the index.
/opt/software/bowtie2-2.1.0/bowtie2-build RefSeq_data.fasta RefSeq_data.fasta _Index

Step 3: Align the transcriptome with reference genome using bowtie

The second step in our pipeline is to align the paired end reads to the reference genome. We are using the software bowtie2, which was created to align short read sequences to long sequences such as the scaffolds in a reference assembly. It requires at least two input files, a FASTQ file containing raw sequence data and a reference genome file in FASTA format.

/opt/software/bowtie2-2.1.0/bowtie2-align -x RefSeq_data.fasta _Index SRR1772683_trimmed.fastq -S sample.sam

Step 4: Sam to bam conversion
samtools view -bS -o sample.bam sample.sam (input .sam file)
samtools sort –O 'bam' sample.bam sample-sorted

Step 5: Merge two sample bam files (optional)
samtools merge -O bam -@ 100 16_24_brown_merged.bam 16_sorted.bam 24_sorted.bam

Step 6: **Identify Variants (app: *Calling SNPs INDELs with SAMtools BCFtools*)**
samtools faidx RefSeq_data.fasta

Step 7: Run 'mpileup' to generate VCF format
samtools mpileup -g -f my.fasta sample-sorted.bam -o sample.bcf

Step 8: Verify Variants (app: SAMTOOLS-0.1.19 VCF-Utilities varFilter)
Not all variants that we called are necessarily of good quality, so it is essential to have a quality filter step. The VCF includes several fields with quality information. The most obvious is the column QUAL, which gives us a Phred-scale quality score.
bcftools filter --exclude 'QUAL<30' sample_variants.vcf |bcftools view -g ^miss > filtered_variants.vcf

GATK (Genome Analysis Toolkit)

(Variant Discovery in High Throughput Sequencing Data)

Source link: https://software.broadinstitute.org/gatk/
Requirements: A Unix-style OS and Java 1.8, bcftools, bowtie, Picard tool.
java -jar /opt/software/new1/GenomeAnalysisTK.jar --help
java -jar /opt/software/new1/GenomeAnalysisTK.jar -T UnifiedGenotyper -R GCF_001704415.1_ARS1_genomic.fasta (reference file) -I /SNP_analysis/set1_13_23/13test_RGadded.bam (input bam file) -o 13_white_unifiedGenotyper_snps.raw.vcf

Step1:

java -jar /opt/software/picard.jar AddOrReplaceReadGroups I=44_47_F.bam RGID=Flower RGLB=Flower RGPL=Illumina RGPU=run_barcode RGSM=P_81 O=44_47_F_picard.bam

Step 2: Make dictionary of reference genome

java -jar /opt/software/picard.jar CreateSequenceDictionary R=/Backup/rao1/sarika/CG_transcriptome_Trinity.fasta O=CG_transcriptome_Trinity.dict

Step 3: Make index file of reference genome

opt/software/applications/samtools-1.2/samtools faidx /Backup/rao1/sarika/CG_transcriptome_Trinity.fasta

Step 4 : reorder bam file

java -jar /opt/software/picard.jar ReorderSam I=44_47_F_picard.bam O=44_47_F_picard_reordered.bam R=/Backup/rao1/sarika/CG_transcriptome_Trinity.fasta CREATE_INDEX=TRUE

Step 5 : Sorting of bam file

java -jar /opt/software/picard.jar SortSam SORT_ORDER=coordinate I=44_47_F_picard_reordered.bam O=44_47_F_picard_reordered_sorted.bam

Step 6 : Indexing of bam file

java -jar /opt/software/picard.jar BuildBamIndex I=44_47_F_picard_reordered_sorted.bam

Sep 7 : Variant Discovery , run GATK

java -jar /opt/software/new1/GenomeAnalysisTK.jar -T UnifiedGenotyper -R /Backup/rao1/sarika/CG_transcriptome_Trinity.fasta -I 44_47_F_picard_reordered_sorted.bam --filter_reads_with_N_cigar -o 44_47_F_picard_reordered_sorted.bam.vcf

Step 8 : Variant Statistics , run GATK

opt/software/bcftools-1.6/bcftools stats SRR1772681_RGadded_reordered_sorted.vcf > SNP_stats.txt

Establishing Marker-QTL Linkage: Principles, Requirements and Methodologies

The idea of using genetic markers to locate the individual quantitative trait locus (QTL) responsible for variation in quantitative traits goes back nearly to the beginning of modern genetics (Sax, 1923). With the availability of dense highly informative marker maps, it has recently become feasible to map genes or QTL accounting for part of the heritability of continuously distributed traits in experimental crosses as well as outbred populations. The most extensive comparative data set available at this point probably comes from QTL mapping efforts in plants. Interestingly, an unexpectedly high proportion of QTL affecting seed size, height, flowering and other complex traits do correspond among different taxa (Paterson, 1998). The process of QTL analysis requires 1) a suitable mapping population of phenotypically contrasting parents, 2) a linkage map of molecular markers, 3) mapping methods and software and 4) reliable phenotypic screening methods and generation of phenotypic data.

Suitable mapping population

It would be always advantageous using populations of early generations such as F_2, F_3, BC population *etc*, since these populations are amenable to make accurate predictions. However, the predictions made involving early generations would be misleading because of camouflaging effect in early generation of the major gene on many other minor genes. Continuous inbreeding to evolve recombinant inbred lines (RILs) can eliminate this camouflaging effect (Allard and Harding, 1963). Thus, RILs can remain as the best choice of population for QTL analysis. As an alternative doubled haploid (DH) lines can also be used. The inherent homozygosity prevailing in the individuals of these two populations make the RILs and DHLs as immortals and help to have as many replications as required by the experiment.

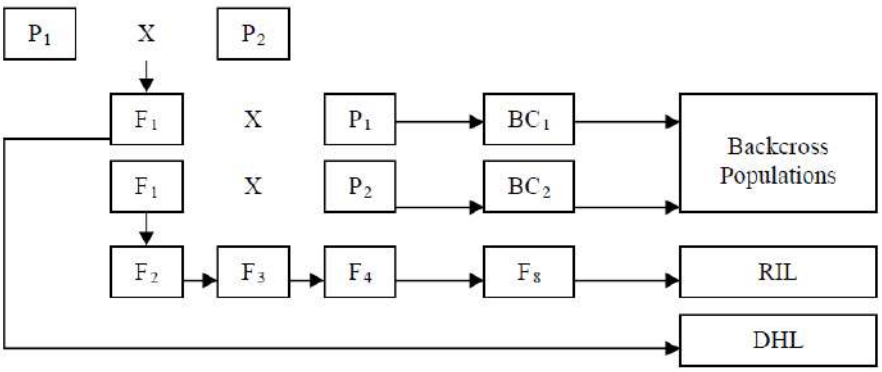


Fig 1. Mapping populations for QTL mapping

A linkage map of molecular markers

Thoday (1961) emphasized that the main practical limitation in localizing QTL, seems to be the non-availability of suitable markers. This limitation was remedied by the construction of complete Restriction Fragment Length Polymorphism (RFLP) linkage maps, permitting systematic searches of an entire genome for QTL influencing a trait (Paterson *et al* 1988). The Amplified Fragment Length Polymorphism (AFLP) markers, (Vos *et al.*, 1995), the markers of choice, remain the best alternative to construct the linkage maps in a very short period based on the existing RFLP maps (Maheswaran *et al*., 1997). Several linkage maps of molecular markers have been constructed exclusively for QTL analysis of various agronomic traits in crop species such as tomato, maize, rice and soybean.

Mapping methods and software

The basis of all QTL detection, regardless of the crop to which it is applied, is the identification of association between genetically determined phenotypes and specific genetic markers. The possible methods of analysis to detect QTL include: 1) single marker analysis (otherwise called as Marker-Trait (MT) Method) and 2) interval analysis.

QTL mapping methods

Conceptually, QTL mapping amounts to a three-step recipe: scan the entire genome with a dense collection of genetic markers; calculate an appropriate linkage statistic $S(x)$ at each position x along the genome; and identify the regions in which the statistic S shows a significant deviation from what would be expected under independent assortment. The underlying assumptions of QTL mapping involving molecular markers are : 1) genes controlling quantitative traits are located on the genome, just like simple genetic markers, 2) if the markers cover a large portion of the genome then there is a large chance that some of the genes controlling the quantitative traits are linked with some of the genetic markers and 3) if the genes and markers are segregating in a genetically defined population, then the linkage relationship among them may be resolved by studying the association between trait variation and marker segregation pattern. The association between quantitative trait variation and marker segregation pattern can be carried out by the following methods.

Single marker analysis

The single marker analysis (SMA) is a good start not only for learning QTL mapping, but also for practical data analysis. Single marker analysis is the method used in earliest studies on QTL mapping (Edwards *et al.*, 1987; Weller *et al.*, 1988). In this, one marker is involved at a time to find the QTL-marker association. The single marker analysis can be implemented as a simple t-test, ANOVA, linear regression, and likelihood ratio test and maximum likelihood estimation (Haley and Knott, 1992; Nienhuis *et al.*, 1987; Wang *et al.*, 1994).

SMA is simple in terms of data analysis and implementation. It can be performed using common statistical software. Gene orders and complete linkage map are not required.

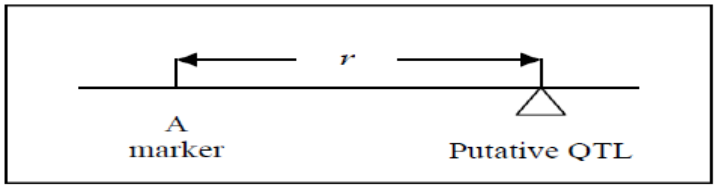


Figure 2: Association of a marker with a putative QTL

The disadvantages of the single marker analysis are: 1) the putative QTL genotypic means and QTL positions are confounded. These confounding cause the estimated QTL effects to be biased and the statistical power to be low particularly when linkage map density is low and 2) QTL positions cannot be precisely determined, due to the non-dependence among the hypothesis tests for linked markers that confound QTL effect and position. Worked out examples to do single marker analysis to establish marker-QTL association are given by Liu, (1998) and some of the key references for single marker analysis are given below.

The disadvantages of the single marker analysis are: 1) the putative QTL genotypic means and QTL positions are confounded. These confounding cause the estimated QTL effects to be biased and the statistical power to be low particularly when linkage map density is low and 2) QTL positions cannot be precisely determined, due to the non-dependence among the hypothesis tests for linked markers that confound QTL effect and position. Worked out examples to do single marker analysis to establish marker-QTL association are given by Liu, (1998) and some of the key references for single marker analysis are given below.

Table 1: Methods to carry out Single Maker Analysis

Method	References
ANOVA	Edwards *et al* (1987)
Simple t-test	Tanksley and Hewitt (1988)
Linear regression	Haley and Knott (1992); Martinez and Curnow (1992), Jansen (1993)
Likelihood ratio	Weller (1986)
Maximum likelihood estimation	Lander and Botstein (1989); Jansen (1992); Luo and Kearsey (1992) Carbonell *et al*. (1992)

Interval analysis or Interval mapping

Interval mapping (IM) is considered as a second level of QTL mapping. QTL mapping by this method requires prior construction of a marker genetic map. The interval mapping

approach is based on the joint frequencies of a pair of adjacent markers and a putative QTL flanked by the two markers.

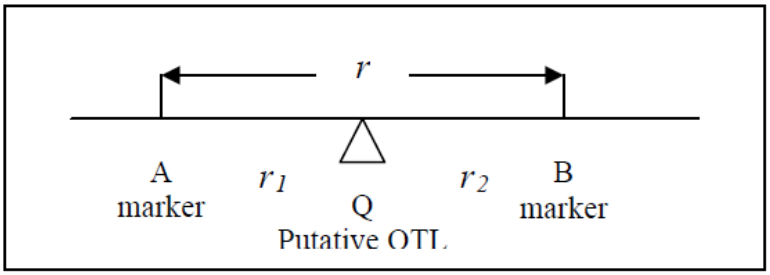


Figure 3: Association of a putative QTL to two flanking markers

Interval mapping can be done by the following methods.

Table 2: Methods employed in Interval Analysis

Method	References
Likelihood approach	Lander and Botstein (1989)
Regression approach	Knapp *et al.* (1990)
Combination of likelihood and regression approach	Zeng (1994)

The approach of interval mapping (IM), otherwise called as simple interval mapping (SIM) considers one QTL at a time in the model for QTL mapping. Therefore, SIM can bias identification and estimation of QTL when multiple QTL are located in the same linkage group (Zeng, 1994). SIM evaluates the association between the trait values and the expected contribution of hypothetical QTL (target QTL) at multiple analysis points between each pair of adjacent marker loci (the target interval). The expected QTL genotype is estimated from the genotypes of flanking marker loci and their distance from the QTL. Since there is usually uncertainty in the QTL genotype, the likelihood is sum of terms, one for each possible QTL genotype, weighted by the probability of that genotype given the genotypes of the flanking markers. The analysis point that yields the most significant association may be taken as the location of a putative QTL. Although IM represented a significant contribution to QTL analyses, it is based on the null hypothesis of no QTL: an incorrect assumption for quantitative traits.

Multiple QTL Mapping

Both SMA and IM are biased when multiple QTL are linked to the marker/interval being considered. To deal with multiple QTL problems, Jansen (1993) Rodolphe and Lefort (1993) and Zeng (1993) independently proposed the idea of combining SIM with multiple regression analysis in mapping. Multiple regression methods were integrated with IM to increase the probability of including all significant QTL in the model. This method was named as

composite interval mapping (CIM). Though CIM produced more accurate and precise estimates than IM, the inclusion of too many cofactors reduced the power to identify QTL relative to IM (Zeng, 1994; Utz and Melchinger, 1996). Kao *et al.* (1999) proposed new method *viz*, multiple interval mapping (MIM) to deal with the mapping of multiple QTL. When compared to SIM and CIM, MIM tends to be more powerful and precise in detecting QTL.

Composite Interval Mapping (CIM)

CIM evaluates the possibility of a target QTL at multiple analysis points across each intermarker interval. However, at each point it also includes the effect of one or more background markers, as defined in SIM. The inclusion of a background marker in the analysis helps in one of two ways, depending on whether the background marker and the target interval are linked. If they are not linked, inclusion of the background marker makes the analysis more sensitive to the presence of a QTL in the target interval. If they are linked, inclusion of the background marker may help to separate target QTL from other linked QTL on the far side of the background marker (Zeng, 1993, 1994).

Multiple Interval Mapping (MIM)

MIM method uses multiple marker intervals simultaneously to fit multiple putative QTL directly in the model for mapping QTL. The MIM method is based on Cockerham's model for interpreting genetic parameters and the method of maximum likelihood for estimating genetic parameters. With the MIM approach, the precision and power of QTL mapping could be improved. Also, epistasis between QTL, genotypic values of individuals and heritabilities of quantitative traits can be readily estimated and analyzed.

Power, precision and accuracy of QTL mapping

QTL analysis includes three stages: detection, mapping and fine mapping. Detection and mapping (estimating a chromosomal location) are often accomplished simultaneously, but they are logically and statistically distinct (Beavis, 1998).

Power of detection

Power is the probability of identifying a QTL of known magnitude, given the predetermined frequency of false positive association (p). Each QTL detection experiment provides an estimate of the strength of a QTL. In some experiments, the QTL will be over estimated, in others, underestimated. This variability may determine whether the QTL appears to be statistically significant, that is, whether the QTL is detected in that experiment. The power of a QTL detection experiment, at a given level of statistical significance depends upon the strength of the QTL and the number of progeny in the population.

The strength of the QTL can be determined based on the fraction of the total trait variance that it explains. Those, which explain over 20 percent of the variance, are strong QTL; traits controlled by such QTL can be considered almost Mendelian. At the other extreme, weak QTL, which explain 1 percent or less of the trait variance, require at least a thousand progeny to detect them with high power. Detection of such QTL is routinely feasible. Between those extremes are moderate QTL, which can be detected with crosses of reasonable size but not necessarily at high power.

Precision of mapping

Precision is a measure of the dispersion of repeated independent estimates of genomic positions or genetic effects of the alleles at QTL and reported by inverse measures such as standard errors or confidence intervals. The size of a confidence interval is expected to be inversely proportional to the number of progeny in the mapping population and to the square of the strength of the QTL. Weak QTL, as defined above, can be assigned to a chromosome, but not located with more precision. Strong QTL can be located by a large backcross or intercross in confidence intervals as small as 11 cM. For strong QTL, precision is limited by the number of recombinants in backcross or intercrosses and QTL can be located more precisely in advanced intercrosses (Darvasi, 1998).

The power and precision of QTL mapping depends on the test statistic derived based on the asymptotic distribution. When single marker analysis is involved in QTL mapping, ‘t’ and ‘F’ statistics are used to assess the power and precision of contrasting marker genotypic classes (Soller *et al*., 1976; McMillan and Robertson, 1974).

Accuracy of mapping

Accuracy is a measure of how close the estimates are to the true values. In practice, acuracy is very difficult to estimate for experimental results because the true values are unknown.

Test statistic for claiming QTL detection

The QTL, by definition, are merely significant statistical associations. These significant associations are detected by having suitable test statistic, otherwise called as ‘critical value’ or ‘threshold statistic’. The reliability and efficiency of the QTL mapping depend considerably on the validity and relevance of the statistical tests used to detect the presence of QTL. Clear statistical guidelines for the interpretation of linkage results are needed to avoid a flood of false positive (presence of a QTL when actually it is not present) claims. At the same time, an overly cautions approach runs the risk of causing true hints of linkage to be missed (false negative).

‘Critical value’ or ‘threshold statistic’ is a limit fixed to eliminate the detection of spurious QTL and QTL with smaller effects. Fixing a suitable threshold statistic for each population

size will help in improving the power of QTL mapping. In using single marker analysis, test of significance is used as threshold statistic. When analysis of variance is used as method to detect QTL, 'F' value is used as threshold statistic. In the same manner, for interval mapping LOD score is used as a threshold statistic. The LOD score summarizes the strength of evidence in favour of the existence of QTL with an effect at a position; if the LOD score exceeds a predetermined threshold (usually LOD score of 3.0 is fixed), the presence of a QTL is inferred. For estimating the LOD score, one has to have the odds ratio, which is the ratio between chance of QTL at a given site and chance of no QTL at a given site.

Threshold statistic adopted for different methods may not have the same strength, resulting in differences in detecting the QTL. Under the circumstance, the threshold statistic of each method has to be evolved to eliminate the discrepancies between methods. Churchill and Deorge (1994) evolved a method to relate the LOD score and F statistic of ANOVA.

$$LOD = [(n_1 + n_2)/2] \log 10 [1+ (T^2/ n_1 + n_2-2)]$$

where, T^2 is F statistic of ANOVA and $(n_1 + n_2)$ is sample size.

When using any statistic of any method, as a criterion in model selection for QTL detection, it is very important to determine the appropriate critical value or threshold value for claiming QTL detection such that correct statistical interference about QTL parameters can be made.

Lander and Botsterin (1989) suggested using the Bonferroni argument for the sparse map case and Orenstein-uhlenback diffusion for the dense map case to determine the critical value. Generally, it has been pointed out that the critical value might need to be adjusted for the number and size of interval, different levels of heritability, different number of multiple linked or unlinked and unlinked in the same or opposite direction (Lander and Botstein, 1989; Jansen, 1993; Zeng, 1994). Visscher and Haley (1996) suggested that the critical value should be reduced after a QTL of large effect has been detected. However, most of this information is not available before mapping and consequently the answers to most of the above questions remain unknown. Churchill and Deorge (1994) therefore suggested using permutation test for determining an appropriate critical value for specific data sets.

The permutation test (Churchill and Doerge, 1994; Deorge and Churchill, 1996) is a method for establishing the significance of the LRS generated by single locus association or interval mapping. In this test, the trait values are randomly permitted among the progeny, destroying the relationship between the trait values and the genotypes of the marker loci in the observed data, QTL parameters and LRS value are estimated for each permuted data set at regular intervals throughout the genome (or some part of the genome) and the maximum LRS is recorded. This procedure is repeated numerous times, giving a distribution of LRS values expected if there were no QTL linked to any of the marker loci. An empirical *p*-value can be obtained for a given LRS by computing the proportion of permuted data sets for which LRS

exceeds the LRS for the observed data. Alternatively, values at appropriate percentile points of the empirical distribution can be used as LRS threshold values to establish significance of the observed LRS. For example, the 95th percentile value is that which would establish significance corresponding to the usual criterion of p = 0.05. Churchill and Doerge (1994) recommended at least 1000 permutations for establishing a threshold for p = 0.05. Permutation tests, therefore, can be time consuming and may be impractical on some computers.

Fixing a correct critical value to detect QTL is still a debatable issue in QTL mapping. Having a uniform stringent standard such as a critical value based on a whole genome search or a critical value based on an infinitely dense map is not acceptable since some times QTL mapping involves few markers (or few chromosomal regions) or a sparse map. Under the circumstance, a hierarchical search – in which one performs a genome scan with a sparse map and then follows up interesting regions with a denser map as suggested by Lander and Kruglyak (1999) is an efficient study design.

Lander and Kruglyak (1995) proposed the following classification based on the number of times that one would expect to see a result at random in a dense, complete genome scan:

Suggestive linkage: Statistical evidence that would be expected to occur one time at random in a genome scan.

Significant linkage: statistical evidence expected to occur 0.05 times in a genome scan (that is, with probability 5 percent).

Highly significant linkage: statistical evidence expected to occur 0.001 time in a genome scan.

Confirmed linkage: significant linkage from one or a combination of initial studies that has subsequently been confirmed in a further sample, preferably by an independent group of investigators. For confirmation, a nominal p value of 0.01 should be required.

Software

Compared to general statistical analysis of biological data, statistical analysis for the study of genes controlling complex traits has the following characteristics: 1) many repeated analysis in one task, 2) lack of standard distribution for some test statistics and 3) complexity of models used in QTL mapping. For using these software packages a known linkage map is needed for either running the programmes or interpreting results. Several companion packages are also available for linkage map construction.

These packages have some similarities such as: 1) interface is not user friendly compared to some commercial software, 2) user support is also limited due to their non-commercial status,

3) statistical models which can be built using the software are limited and 4) speed of model building is high for the models which the software can build. The details on some of the software routinely employed in QTL mapping are given below.

MAPMAKER/QTL (Lincoln *et al*., 1992b) is a widely used program for UNIX or DOS operating systems and is the original QTL mapping program intended for distribution. It can perform composite interval mapping, although the documentation does not use that term; but it cannot perform permutation tests. It requires the companion program MAPMAKER/EXP (Lander *et al*., 1987; Lincoln *et al*., 1992a) to format data and to calculate marker maps.

QTL Cartographer (Basten *et al*. 1994, 1997) is a suite of programs for DOS, UNIX, or Mac OS. They are designed to be used in sequence, each accepting input in the form of text files and storing its output in text files for the next program. This suite offers several variations of CIM with automatic selection of background loci. It also has provision for estimating confidence intervals by resampling. QTL Cartographer, MapQTL, and PLABQTL are similar in many respects. QTL Cartographer is distinguished by its menu-driven interface, more detailed documentation, resampling methods and the lack of a licensing fee.

Map Manager QT (Manly and Elliott, 1991; Manly, 1997) is a program for Mac OS distinguished by its graphical user interface for data entry, editing, manipulation, and display. It is designed to be used either as a mapping program itself or as a data-preparation program for other mapping programs.

QGene (Nelson, 1997) is a commercial program for Mac OS whose strength is a variety of graphics for displaying trait data and relationships among marker genotypes and between traits and marker genotypes. These functions make it uniquely useful for rapid exploration of data. However, it does not perform CIM.

MapQTL (van Ooijen and Maliepaard, 1996) is a commercial program for several operating systems that is distinguished by its ability to map QTL in populations derived from non-inbred parents, in which both markers and QTL may have more than two alleles. It also offers a nonparametric form of single-locus association, the Kruskal-Wallis rank sum test, appropriate for data with distributions far from normal.

PLABQTL (Utz and Melchinger, 1996) is a script-driven program for DOS or AIX that is designed to analyze automatically a dataset at increasing levels of complexity in successive runs. The final level is capable of evaluating the effect of different environments and the effect of interactions between QTL and environmental effects.

MQTL (Tinker and Mather, 1995a, 1995b) is a program for DOS or Sun OS that uses a simplified form of composite interval mapping (sCIM) for mapping QTL in large data sets

derived from multiple environments. Like PLABQTL, it will estimate environmental effects and QTL-environment interactions.

Multimapper (Sillanpaa, 1998) is a program for UNIX that implements a Bayesian method for building multi-QTL models automatically. Multimapper is designed to map QTL within a single linkage group and it produces a plot of QTL probability as a function of map distance. This type of plot seems intuitively more interpretable than the plot of the likelihood ratio statistic or LOD score produced by other programs. However, it seems to be the most suited to the analysis of single chromosomes for which other programs have indicated the possibility of multiple QTL. Multimapper is designed to work with QTL Cartographer as a companion program.

Epistat (Chase *et al*., 1997) is a program for DOS designed primarily for the detection and analysis of interactions between QTL. It does not perform interval mapping and therefore does not require mapped markers. It is an interactive program, displaying graphic results in response to single-keystroke commands.

The QTL Cafe is a program being developed in Java to make it available for multiple computer platforms. It is currently available as an applet that runs in a Java-enabled World Wide Web browser.

Available public domain software packages for studying genes controlling complex traits are not adequate for development of genomic research on complex traits in terms of user interface, flexibility and user support. Software packages with commercial quality are needed to accommodate the growing needs of data analysis and management in genomic research. This is especially true for study of genes controlling complex traits.

Reliable phenotypic screening and generation of phenotypic data

To adequately explore the QTL during the mapping phase, the phenotype must be evaluated in replicated trials in different environments. Moreover, phenotypic screening should be done based on reliable and reproducible screening methods. Large data sets can be generated by the coordinated efforts of several groups, providing valuable information about genes governing quantitative characters in a range of environments. Such data will provide information about the magnitude of the effect of different QTL and whether there is interaction between QTL and environment.

The issues related to population development and construction of linkage maps do not pose many problems with the existing level of knowledge. Though, issues related to methods for detecting QTL, software for QTL analysis are having problems such as inaccurate detection of QTL (occurrence of false positives and false negatives), the issues associated with phenotypic screening pose severe threat to an emerging tool of plant breeding.

Most agronomically important characters involve multiple genes that interact with each other and with the environment in complex ways. This creates a situation wherein QTL can be detected only some of the time. This necessitates designing of experiments to qualify the value of specific QTL. To adequately explore the value of QTL, the phenotypes must be evaluated in well-replicated trials in different environments. The conduct of replicated trials warrants a suitable population which can be effectively replicated. Here comes the problem of developing an immortal population such as RILs and DHLs. Developing both kinds of populations and their maintenance is a cumbersome process even by a well established breeding institute. QTL are hypothetical genes based on statistical inference. Genetic effects used in QTL mapping could have very little biological meaning. To have a biological meaning of QTL mapping the selection of traits to be phenotyped is very important. More over, phenotype of several traits is not amenable for QTL dissection.

Conclusion

QTL, otherwise described as hypothetical genes based on statistical inferences, have very little biological meaning. To date the knowledge on QTL mapping is enormous. However, the accrued knowledge does not have immediate solutions to the problems associated with QTL mapping. Some of the approaches such as adopting metabolic genetic model (Byrne *et al*., 1996; Mitchell-Olds and Pedersen, 1998) and candidate gene concept (Long and Longly, 1999) in conjunction with Single Nucleotide Polymorphism (SNP) may make the QTL mapping approach as biologically meaningful one. Issues related to genetic mapping of QTL are well reviewed by Liu (1998), Lynch and Walsh (1997) and Paterson (1998).

References

Allard RW and Harding J (1963). Early generation analysis and prediction of gain under selection in derivatives of a wheat hybrid. Crop Sci. 3: 454 – 456.

Basten C, Weir BS and Zeng Z-B (1994). Z map – QTL Cartographer. pp 65 – 66. In: Proceedings of the 5th World Congress on Genetics Applied to Livestock Production. Computing strategies and software. (Eds) Smith C, Gavora JS, Benkel B, Chesnais J, Fairfull W, Gipson JP, Kennedy BW and Burnside EB. Vol 22.

Basten C, Weir BS and Zeng Z-B (1997). QTL Cartographer. A reference manual and tutorial for QTL mapping (Raleigh. N.C: Department of Statistics. North Carolina State University).

Beavis WD (1998). QTL analyses: Power, precision and accuracy. Pp 145 – 162. In: Molecular dissection of complex traits. Paterson AH (Ed), CRC Press. Boca Raton, New York.

Byrne PF, McMuller MD, Snook ME, Musket TA, Theuri JM, Widstrom NW, Wisemen BR and Coe EH (1996). Quantitative trait loci and metabolic pathways: Genetic control of the concentration of maysin, a corn earworm resistance factor in maize silks. Proc. Natl. Acd. Sci. USA. 93: 8820-8825.

Carbonell EA, Greig TM, Balansard E and Asians MJ (1992). Interval mapping in the analysis of non-additive quantitative trait loci. Biometrics 48: 305–315.

Chase K, Adler FR and Lark KG (1997). Epistat : a computer programme for identifying and testing interactions between pairs of quantitative trait loci. Theor. Appl. Genet. 94: 724 – 730.

Churchill GA and Deorge RW (1994). Empirical threshold values for quantitative trait mapping. Genetics 138: 963 – 971.

Darvasi A (1998). Experimental strategies for the genetic dissection of complex traits in animal models. Nat. Genet. 18: 19 – 24.

Doerge RW and Churchill GA (1996). Permutation tests for multiple loci affecting a quantitative character. Genetics 142: 285 – 294

Edwards MD, Stuber CW and Wendel JF (1987). Molecular marker facilitated investigations of quantitative trait loci in maize I. Numbers, genomic distribution and types of gene action. Genetics 116 : 113 – 125.

Haley CS and Knott SA (1992). A simple regression method for mapping quantitative trait loci in line crosses using flanking markers. Heredity 69: 315 –324.

Jansen RC (1992). A general mixture model for mapping quantitative trait loci by using molecular markers. Theor. Appl. Genet. 85: 252 – 260.

Jansen RC (1993). Interval mapping of multiple quantitative trait loci. Genetics 135: 205 – 211.

Kao CH, Zeng Z-B and Teasdale RD (1999). Multiple interval mapping. Genetics 152: 1203 – 1216.

Knapp SJ, Bridges WC.Jr and Brikes D (1990). Mapping quantitative trait loci using molecular marker linkage maps. Theor. Appl. Genet. 79: 583 – 592.

Lander ES and Botstein D (1989). Mapping Mendelian factors underlying quantitative traits using RFLP linkage maps. Genetics 139: 1421 – 1428.

Lander ES and Kruglyak L (1995). Genetic dissection of complex traits: guidelines for interpreting and reporting linkage results. Nat. Genet. 11: 241 – 247.

Lander ES, Green P, Abrahamson J, Barlow A, Daly M, Lincoln SE and Newburg L (1987). Mapmaker: an interactive computer package for constructing primary genetic linkage maps of experimental and natural populations. Genomics 1: 174-181.

Lincoln S, Daly M and Lander E (1992a). Constructing linkage map with MAPMARKER/EXP. Whitehead Institute Technical Report.

Lincoln S, Daly M and Lander E (1992b). Mapping genes controlling quantitative traits with MAPMARKER/QTL. Whitehead Institute Technical Report.

Liu BH (1998). Statistical Genomics: Linkage, mapping and QTL analysis, CRC press, Boca Raton, New York. 611p.

Long AD and Langely CH (1999). The power of association studies to detect the contribution of candidate genetic loci to variation in complex traits. Genome Research 9: 720 – 731.

Luo ZW and Kearsey MJ (1992). Interval mapping of quantitative trait loci in F_2 population. Heredity 69: 236 – 242

Lynch M and Walsh B (1998). Genetics and analysis of quantitative traits. Sunderland, Mass, Sinauer Associates Inc. 980p.

Maheswaran M, Subudhi PK, Nandi S, Xu JC Parco A Yang DC and Huang N (1997). Polymorphism, distribution of AFLP markers in a doubled haploid population. Theor. Appl. Genet. 94: 39-45.

Manly KF (1997). Map manager QT: Software for mapping quantitative trait loci. Abstracts of the 11th International Mouse Genome Conference. St. Petersburg. Fla. pp 75.

Manly KF, Elliott RW (1991). RI manager: A microcomputer program for analysis of data from recombinant inbred strains. Mamm. Genome. 1: 123 – 126.

Martinez O and Curnow RN (1992). Estimating the locations and the sizes of the effects of quantitative trait loci using flanking markers. Theor. Appl. Genet. 85: 480 – 488

McMillan I and Robertson A (1974). The power of methods for the detection of major genes affecting quantitative charcters. Heredity 32: 349-356

Mitchell-Olds T and Pederson D (1998). The molecular basis of quantitative genetic variation in central and secondary metabolism in *Arabodopsis*. Genetics 149: 739-747.

Neinhuis J, Helentjaris T, Slown M, Ruggero B, Schaefer A (1987). Restriction fragment length polymorphism analysis of loci associated with insect resistance in tomato. Crop Sci. 27: 797 – 903.

Nelson JC (1997). Q GENE : Software for marker based genomic analysis and breeding. Mol. Breed. 3: 239 – 245.

Paterson AH, Lander ES, Hewitt. JD, Paterson S, Lincoln SE and Tanksley SD (1988). Resolution of quantitative traits into Mendelion factors, using a complete linkage map of restriction fragment length polymorphisms. Nature 335: 721 – 726.

Rodolphe F and Lefort M (1993). A multiple marker model for detecting chromosomal segments displaying QTL activity. Genetics 134: 1277-1288.

Sax K (1923). The association of size differences with seed coat pattern and pigmentation in *Phaseolus vulgaris*. Genetics 8: 552 – 556.

Sillanppa MJ (1998). Multimapper Reference manual.

Soller M, Brody T and Genizi MA (1976). On the power of experimental designs for the detection of linkage between marker loci and quantitative trait loci in crosses between inbred lines. Theor . Appl. Genet. 47 : 35 – 39.

Tanksley SD and Hewitt JD (!988). Use of molecular markers in breeding for soluble solids in tomato – A re-examination. Theor. Appl. Genet. 75: 811 – 823.

Thoday JM (1961). Location of polygenes. Nature 191: 368 – 370.

Tinker NA and Mather DE (1995b). MQTL : Software for simplified composite interval mapping of QTL in multiple environments. J. Quant. Trait Loci. 1, http://probe.nalusda.gov:8000/otherdocs/jqtl/index.html.

Utz HF and Melchinger AE (1996). PLABQTL: a program for composite interval mapping of QTL. J Quant. Trait Loci . 2.

van Ooijen TW and Maleipaard C (1996). Map QTL™ version 3.D software for the calculation of QTL position on genetic maps (Wageningen : CPRO – DLO).

Visscher PM and Haley CS (1996). Detection of quantitative trait loci in the line cross under infinitesimal genetic models. Theor. Appl. Genet. 93: 691 – 702.

Vos P, Hoger SR, Bleeker M, Lee T, Hornes M, Frijter A, Pot J, Peleman J, Kuiper M and Zabeau M (1995). AFLP: A new technique for finger printing. Nucleic Acids Res. 23: 4407-4414.

Wang GL, Mackill DJ, Bonmann JM, McCouch SR, Champoux MC, and Nelson RJ (1994). RFLP mapping of genes conferring complete and partial resistance to blast in a durably resistant rice cultivar. Genetics 136: 1421-1434.

Weller Ji, Soller M and Bordy T (1988). Linkage analysis of quantitative traits in inter specific cross of tomato (*L. esenlentum* x *L. pimpinellifolium*) by means of genetic markers. Genetics 118: 329 – 339.

Weller JI. (1986). Maximum likelihood techniques for the mapping and analysis of quantitative trait loci with the aid of genetic markers. Biometrics 42: 627-640.

Zeng Z-B (1993). Theoretical basis for seperation of multiple linked gene effects in a mapping quantitative trait loci. Proc. Natl. Acad. Sci. USA. 90 : 10972 – 10976

Zeng Z-B (1994). Precision mapping of quantitative trait loci. Genetics 136: 1457 – 1468.

Planning and Designing of Agricultural Experiments

An experiment is usually associated with a scientific method for testing certain phenomena. An experiment facilitates the study of such phenomena under controlled conditions and thus creating controlled condition is an essential component. Scientists in the biological fields who are involved in research constantly face problems associated with planning, designing and conducting experiments. Basic familiarity and understanding of statistical methods that deal with issues of concern would be helpful in many ways. Researchers who collect data and then look for a statistical technique that would provide valid results will find that there may not be solutions to the problem and that the problem could have been avoided first by a properly designed experiment. Obviously it is important to keep in mind that we cannot draw valid conclusions from poorly planned experiments. Second, the time and cost involved in many experiments are enormous and a poorly designed experiment increases such costs in time and resources. For example, an agronomist who carries out fertilizer experiment knows the time limitation of the experiment. He knows that when seeds are to be planted and harvested. The experimenter plot must include all components of a complete design. Otherwise what is omitted from the experiment will have to be carried out in subsequent trials in the next cropping season or next year. The additional time and expenditure could be minimized by a properly planned experiment that will produce valid results as efficiently as possible. Good experimental designs are products of the technical knowledge of one's field, an understanding of statistical techniques and skill in designing experiments.

Any research endeavor may entail the phases of Conception, Design, Data collection, Analysis and Dissemination. Statistical methodologies can be used to conduct better scientific experiments if they are incorporated into entire scientific process, i.e., From inception of the problem to experimental design, data analysis and interpretation. When planning experiments we must keep in mind that large uncontrolled variations are common occurrences. Experiments are generally undertaken by researchers to compare effects of several conditions on some phenomena or in discovering an unknown effect of particular process. An experiment facilitates the study of such phenomena under controlled conditions. Therefore the creation of controlled condition is the most essential characteristic of experimentation. How we formulate our questions and hypotheses are critical to the experimental procedure that will follow. For example, a crop scientist who plants the same variety of a crop in a field may find variations in yield that are due to periodic variations across a field or to some other factors that the experimenter has no control over. The methodologies used in designing experiments will separate with confidence and accuracy a varietal difference of crops from the uncontrolled variations.

The different concepts in planning of experiment can be well explained through chapati tasting experiment.

Consider an experiment to detect the taste difference in chapati made of wheat flour of c306 and pv 18 varieties. The null hypothesis we can assume here is that there is no taste difference in chapatis made of c306 or pv18 wheat flours. After the null hypothesis is set, we have to fix the level of significance at which we can operate. The pv18 is a much higher yielding variety than c306. Hence a false rejection may not help the country to grow more pv18 and the wheat production may decrease while a false acceptance may give more production of pv18 wheat and the consumption may be less or practically nil. Thus the false acceptance or false rejection are of practically equal consequence and we agree to choose the level of significance at $\alpha = 0.05$. Now to execute the experiment, a subject is to be found with extrasensory powers who can detect the taste differences. The colours of c306 and pv18 are different and anyone, even without tasting the chapatis, can distinguish the chapatis of either kind by a mere glance. Thus the taster of the chapatis has to be blindfolded before the chapatis are given for tasting. Afterwards, the method is to be decided in which the experiment will be conducted. The experiment can be conducted in many ways and of them three methods are discussed here:

- Give the taster equal number of chapatis of either kind informing the taster about it.
- Give the taster pairs of chapatis of each kind informing the taster about it.
- Give the taster chapatis of either kind without providing him with any information. Let us use 6 chapatis in each of these methods.

Under first method of experimentation, if the null hypothesis is true, then the experimenter cannot distinguish the two kinds of chapaties and he will randomly select 3 chapatiS out of 6 chapaties given to him, as made of pvl8 wheat. In that case, all correct guesses are made if selection exactly coincides with the exactly used wheat variety and the probability for such an occurrence is:

$$1/\binom{6}{3} = 1/20 = 0.05$$

Under second method,the pv18 wheat variety chapaties are selected from each pair given if the null hypothesis is true. Furthermore, independent choices are made of pv18 variety chapaties from each pair. Thus the probability of making all correct guesses is

$$1/(2)^3 = 1/8 = 0.125.$$

In third method the experimenter has to make the choice for each chapati and the situation is analogous at calling heads or tails in a coin tossing experiment. The probability of making all correct guesses would then be:

$1/2^6 = 1/64 = .016.$

If the experimenter makes all correct guesses in third method as its probability is smaller than the selected $\alpha = 0.05$, we can reject the null hypothesis and conclude that the two wheat varieties give different tastes at chapaties. In other methods the probability of making all correct guesses does not exceed $\alpha = 0.05$ and hence with either method, we cannot reject the null hypothesis even if all correct guesses are made.

However, if 8 chapaties are used by first method and if the taster guesses all of them, we can reject the null hypothesis, at 0.05 level of significance, as the probability of making all correct guesses would then be $1/\binom{8}{3} = 1/56$ which is smaller than 0.05. 8 chapaties will not enable us to reject the null hypothesis even if all correct guesses are made by second method as the probability of making all correct guesses is $\left(\frac{1}{4}\right)^4 = \frac{1}{16} = 0.06$ it is easy to see that if 10 chapaties are given by second method and if all correct guesses are made, then we can reject the null hypothesis at 0.05 level of significance. Not to unduly influence the taster in making guesses, we should also present the chapaties in a random order rather than systematically presenting them for tasting.

The above discussed chapati tasting experiment brings home the following salient features of experimentation:

- All the extraneous variations in the data should be eliminated or controlled excepting the variations due to the treatments under study. One should not artificially provide circumstances for one treatment to show better results than others.
- Far a given size of the experiment, though the experiment can be done in many ways, even the best results may not turn out to be significant with some designs, while some other design can detect the treatment differences. Thus there is an imperative need the choose the right type of design, before the commencement of the experiment, lest the results may be useless.
- If for some specific reasons related to the nature .of the experiment, a particular method has to be used in experimentation, then adequate number of replications of each treatment have to be provided in order to get valid inferences.
- The treatments have to be randomly allocated to the experimental units.

The terminologies often used in planning and designing of experiments are listed below.

Treatment

Treatment refers to controllable quantitative or qualitative factors imposed at a certain level

by the experimenter. For an agronomist several fertilizer concentrations applied to a particular crop or a variety of crop is a treatment. Similarly, an animal scientist looks upon several concentrations of a drug given to animal species as a treatment. In agribusiness we may look upon impact of advertising strategy on sales a treatment. To an agricultural engineer, different levels of irrigation may constitute a treatment.

Experimental Unit

An experimental unit is an entity that receives a treatment e.g., for an agronomist or horticulturist it may be a plot of a land or batch of seed, for an animal scientist it may be a group of pigs or sheep, for a scientist engaged in forestry research it may be different tree species occurring in an area, and for an agricultural engineer it may be manufactured item. Thus, an experimental unit maybe looked upon as a small subdivision of the experimental material, which receives the treatment.

Experimental Error

Differences in yields arising out of experimental units treated alike are called Experimental Error.

Controllable conditions in an experiment or experimental variable are terms as a factor. For example, a fertilizer, a new feed ration, and a fungicide are all considered as factors. Factors may be qualitative or quantitative and may take a finite number of values or type. Quantitative factors are those described by numerical values on some scale. The rates of application of fertilizer, the quantity of seed sown are examples of quantitative factors. Qualitative factors are those factors that can be distinguished from each other, but not on numerical scale e.g., type of protein in a diet, sex of an animal, genetic make up of plant etc. While choosing factors for any experiment researcher should ask the following questions, like What treatments in the experiment should be related directly to the objectives of the study? Does the experimental technique adopted require the use of additional factors? Can the experimental unit be divided naturally into groups such that the main treatment effects are different for the different groups? What additional factors should one include in the experiment to interact with the main factors and shed light on the factors of direct interest? How desirable is it to deliberately choose experimental units of different types?

Basic Principles of Design of Experiments

Given a set of treatments which can provide information regarding the objective of an experiment, a design for the experiment, defines the size and number of experimental units, the manner in which the treatments are allotted to the units and also appropriate type and grouping of the experimental units. These requirements of a design ensure validity, interpretability and accuracy of the results obtainable from an analysis of the observations.

These purposes are served by the principles of:

- Randomization
- Replication
- Local (Error) control

Randomization

After the treatments and the experimental units are decided the treatments are allotted to the experimental units at random to avoid any type of personal or subjective bias, which may be conscious or unconscious. This ensures validity of the results. It helps to have an objective comparison among the treatments. It also ensures independence of the observations, which is necessary for drawing valid inference from the observations by applying appropriate statistical techniques.

Depending on the nature of the experiment and the experimental units, there are various experimental designs and each design has its own way of randomization. Various speakers while discussing specific designs in the lectures to follow shall discuss the procedure of random allocation separately.

Replication

If a treatment is allotted to r experimental units in an experiment, it is said to be replicated r times. If in a design each of the treatments is replicated r times, the design is said to have r replications. Replication is necessary to

- Provide an estimate of the error variance which is a function of the differences among observations from experimental units under identical treatments.
- Increase the accuracy of estimates of the treatment effects.

Though, more the number of replications the better it is, so far as precision of estimates is concerned, it cannot be increased infinitely as it increases the cost of experimentation. Moreover, due to limited availability of experimental resources too many replications cannot be taken.

The number of replications is, therefore, decided keeping in view the permissible expenditure and the required degree of precision. Sensitivity of statistical methods for drawing inference also depends on the number of replications. Sometimes this criterion is used to decide the number of replications in specific experiments.

Error variance provides a measure of precision of an experiment, the less the error variance the more precision. Once a measure of error variance is available for a set of experimental units, the number of replications needed for a desired level of sensitivity can be obtained as below.

Given a set of treatments an experimenter may not be interested to know if two treatment differ in their effects by less than a certain quantity, say, d. In other words, he wants an experiment that should be able to differentiate two treatments when they differ by d or more.

The significance of the difference between two treatments is tested by t-test where

$$t = \frac{\bar{y}_i - \bar{y}_j}{\sqrt{2s^2/r}},$$

Here, $\bar{y}_i$, and $\bar{y}_j$ are the arithmetic means of two treatment effects each based on r replications, s^2 is measure of error variation.

Given a difference d, between two treatment effects such that any difference greater than d should be brought out as significant by using a design with r replications, the following equation provides a solution of r.

$$t = \frac{|d|}{\sqrt{2s^2/r}},$$

$$r = \frac{t_0^2}{d^2} x 2s^2 \qquad \ldots(1)$$

where t_0 is the critical value of the t-distribution at the desired level of significance, that is, the value of t at 5 or 1 per cent level of significance read from the t-table. If s^2 is known or based on a very large number of observations, made available from some pilot pre-experiment investigation, then t is taken as the normal variate. If s^2 is estimated with n degree of freedom (d.f.) then t_0 corresponds to n d.f.

When the number of replication is r or more as obtained above, then all differences greater than d are expected to be brought out as significant by an experiment when it is conducted on a set of experimental units which has variability of the order of s^2. For example, in an experiment on wheat crop conducted in a seed farm in Bhopal, to study the effect of application of nitrogen and phosphorous on yield a randomized block design with three replications was adopted. There were 11 treatments two of which were (i) 60 Kg/ha of nitrogen (ii) 120 Kg/ha of nitrogen. The average yield figures for these two application of the fertilizer were 1438 and 1592 Kg/ha respectively and it is required that differences of the order of 150 Kg/ha should be brought out significant. The error mean square (s^2) was

12134.88. Assuming that the experimental error will be of the same order in future experiments and t_0 is of the order of 2.00, which is likely as the error d.f. is likely to be more than 30 as there are 11 treatments; Substituting in (1), we get:

$$r = \frac{2t_0^2 s^2}{d^2} = \frac{2x2^2 x2134.88}{150^2} = 4\,(\text{approx.})$$

Thus, an experiment with 4 replications is likely to bring out differences of the order of 150 Kg/ha as significant.

Another criterion for determining r is to take a number of replications which ensures at least 10 d.f. for the estimate of error variance in the analysis of variance of the design concerned since the sensitivity of the experiment will be very much low as the F test (which is used to draw inference in such experiments) is very much unstable below 10 d.f.

Local Control

The consideration in regard to the choice of number of replications ensure reduction of standard error of the estimates of the treatment effect because the standard error of the estimate of a treatment effect is $\sqrt{s^2/r}$, but it cannot reduce the error variance itself. It is, however, possible to devise methods for reducing the error variance. Such measures are called *error control* or local control. One such measure is to make the experimental units homogenous. Another method is to form the units into several homogenous groups, usually called blocks, allowing variation between the groups.

A considerable amount of research work has been done to divide the treatments into suitable groups of experimental units so that the treatment effect can be estimated more precisely Extensive use of combinatorial mathematics has been made for formation of such group treatments. This grouping of experiment units into different groups has led to the development of various designs useful to the experimenter. We now briefly describe the various term used in designing of an experiment

Blocking

It refers to methodologies that form the units into homogeneous or pre-experimental subject-similarity groups. It is a method to reduce the effect of variation in the experimental material on the Error of Treatment of Comparisons. For example, animal scientist may decide to group animals on age, sex, breed or some other factors that he may believe has an influence on characteristic being measured. Effective blocking removes considerable measure of variation nom the experimental error. The selection of source of variability to be used as basis of blocking, block size, block shape and orientation are crucial for blocking. The blocking factor is introduced in the experiment to increase the power of design to detect treatment effects.

The importance of good designing is inseparable from good research (results). The following examples point out the necessity for a good design that will yield good research. First, a nutrition specialist in developing country is interested in determining whether mother's milk is better than powdered milk for children under age one. The nutritionist has compared the growth of children in village A, who are all on mother's milk against the children in village B, who use powdered milk. Obviously, such a comparison ignores the health of the mothers, the sanitary-conditions of the villages, and other factors that may have contributed to the differences observed without any connection to the advantages of mother's milk or the powdered milk on the children. A proper design would require that both mother's milk and the powdered milk be alternatively used in both villages, or some other methodology to make certain that the differences observed are attributable to the type of milk consumed and not to some uncontrollable factor. Second, a crop scientist who is comparing 2 varieties of maize, for instance, would not assign one variety to a location where such factors as sun, shade, unidirectional fertility gradient, and uneven distribution of water would either favor or handicap it over the other. If such a design were to be adopted, the researcher would have difficulty in determining whether the apparent difference in yield was due to variety differences or resulted from such factors as sun, shade, soil fertility of the field, or the distribution of water. These two examples illustrate the type of poorly designed experiments that are to be avoided.

Analysis of Variance

Analysis of Variance (ANOVA) is a technique of partitioning the overall variation in the responses into different assignable sources of variation, some of which are specifiable and others unknown. Total variance in the sample data is partitioned and is expressed as the sum of its non-negative components is a measure of the variation due to some specific independent source or factor or cause. ANOVA consists in estimation of the amount of variation due to each of the independent factors (causes) separately and then comparing these estimates due to ascribable factors (causes) with the estimate due to chance factor the latter being known as experimental error or simply the error.

Total variation present in a set of observable quantities may, under certain circumstances, be partitioned into a number of components associated with the nature of classification of the data. The systematic procedure for achieving this is called *Analysis of Variance*. The initial techniques of the analysis of variance were developed by the statistician and geneticist R. A. Fisher in the 1920s and 1930s, and is sometimes known as Fisher's analysis of variance, due to the use of Fisher's F-distribution as part of the test of statistical significance.

Thus, ANOVA is a statistical technique that can be used to evaluate whether there are differences between the average value, or mean, across several population groups. With this model, the *response variable is continuous* in nature, whereas the *predictor variables are*

categorical. For example, in a clinical trial of hypertensive patients, ANOVA methods could be used to compare the effectiveness of three different drugs in lowering blood pressure. Alternatively, ANOVA could be used to determine whether infant birth weight is significantly different among mothers who smoked during pregnancy relative to those who did not. In a particular case, where two population means are being compared, ANOVA is equivalent to the independent two-sample *t*-test.

The fixed-effects model of ANOVA applies to situations in which the experimenter applies several treatments to the subjects of the experiment to see if the response variable values change. This allows the experimenter to estimate the ranges of response variable values that the treatment would generate in the population as a whole. In it factors are fixed and are attributable to a finite set of levels of factor eg. Sex, year, variety, fertilizer etc.

Consider for example a clinical trial where three drugs are administered on a group of men and women some of whom are married and some are unmarried. The three classifications of sex, drug and marital status that identify the source of each datum are known as factors. The individual classification of each factor is known as levels of the factors. Thus, in this example there are 3 levels of factor drug, 2 levels of factor sex and 2 levels of marital status. Here all the effects are fixed. Random effects models are used when the treatments are not fixed. This occurs when the various treatments (also known as factor levels) are sampled from a larger population. When factors are random, these are generally attributable to infinite set of levels of a factor of which a random sample are deemed to occur *eg*. research stations, clinics in Delhi, sire, etc. Suppose new inject-able insulin is to be tested using 15 different clinics of Delhi state. It is reasonable to assume that these clinics are random sample from a population of clinics from Delhi. It describe the situations where both fixed and random effects are present.

In any ANOVA model, general mean is always taken as fixed effect and error is always taken as random effect. Thus class of model can be classified on the basis of factors, other than these two factors. ANOVA can be viewed as a generalization of *t*-tests: a comparison of differences of means across more than two groups.

The ANOVA is valid under certain assumptions. These assumptions are:

- Samples have been drawn from the populations that are normally distributed.
- Observations are independent and are distributed normally with mean zero and variance σ^2.
- Effects are additive in nature.

The ANOVA is performed as one-way, two-way, three-way, etc. ANOVA when the number of factors is one, two or three respectively. In general if the number of factors is more, it is termed as multi-way ANOVA.

In this chapter, three basic designs viz., Completely randomized design (CRD), Randomized Complete Block Design (RCBD) and Latin Square Design (LSD) are explained in detail.

Completely Randomized Design

Designs are usually characterized by the nature of grouping of experimental units and the procedure of random allocation of treatments to the experimental units. In a completely randomized design the units are taken in a single group. As far as possible the units forming the group are homogeneous. This is a design in which only randomization and replication are used. There is no use of local control here.

Let there be *v* treatments in an experiment and *n* homogeneous experimental units. Let the i^{th} treatment be replicated r_i times *(i = 1,2,..., v)* such that $\sum_{i=1}^{v} r_i = n$. The treatments are allotted at random to the units.

Normally the number of replications for different treatments should be equal as it ensures equal precision of estimates of the treatment effects. The actual number of replications is, however, determined by the availability of experimental resources and the requirement of precision and sensitivity of comparisons. If the experimental material for some treatments is available in limited quantities, the numbers of their replication are reduced. If the estimates of certain treatment effects are required with more precision, the numbers of their replication are increased.

Randomization

There are several methods of random allocation of treatments to the experimental units. The *v* treatments are first numbered in any order from *1* to *v*. The *n* experimental units are also numbered suitably. One of the methods uses the random number tables. Any page of a random number table is taken. If *v* is a one-digit number, then the table is consulted digit by digit. If *v* is a two-digit number, then two-digit random numbers are consulted. All numbers greater than *v* including zero are ignored.

Let the first number chosen be n_1; then the treatment numbered n_1 is allotted to the first unit. If the second number is n_2 which may or may not be equal to n_1 then the treatment numbered n_2 is allotted to the second unit. This procedure is continued. When the i^{th} treatment number has occurred r_i times, $(i = 1,2,...,v)$ this treatment is ignored subsequently. This process terminates when all the units are exhausted.

One drawback of the above procedure is that sometimes a very large number of random numbers may have to be ignored because they are greater than *v*. It may even happen that the random number table is exhausted before the allocation is complete. To avoid this difficulty the following procedure is adopted. We have described the procedure by taking *v* to be a two-digit number. Let *P* be the highest two-digit number divisible by *v*. Then all numbers greater than *P* and zero are ignored. If a selected random number is less than *v*, then it is used as such. If it is greater than or equal to *v*, then it is divided by *v* and the remainder is taken to the random number. When a number is completely divisible by *v*, then the random number is *v*. If *v* is an *n*-digit number, then *P* is taken to be the highest *n*-digit number divisible by *v*. The rest of the procedure is the same as above.

Analysis

This design provides a one-way classified data according to levels of a single factor. For its analysis the following model is taken:

$$y_{ij} = \mu + t_i + e_{ij}\ , \qquad i = 1,\cdots,v; j = 1,\cdots r_i,$$

where y_{ij} is the random variable corresponding to the observation y_{ij} obtained from the j^{th} replicate of the i^{th} treatment, μ is the general mean, t_i is the fixed effect of the i^{th} treatment and e_{ij} is the error component which is a random variable assumed to be normally and independently distributed with zero means and a constant variance σ^2.

Let $\sum_j y_{ij} = T_i \quad (i = 1,2,...,v)$ be the total of observations from i^{th} treatment. Let further $\sum_i T_i = G$. Correction factor (*C.F.*) $= G^2/n$.

Sum of squares due to treatments $= \sum_{i=1}^{v} \frac{T_i^2}{r_i} - C.F.$

Total sum of squares $= \sum_{i=1}^{v}\sum_{j=1}^{r_i} y_{ij}^2 - C.F.$

ANALYSIS OF VARIANCE

Sources of variation	**Degrees of freedom (D.F.)**	**Sum of squares (S.S.)**	**Mean squares (M.S.)**	**F**
Treatments	*v – 1*	*SST* $= \sum_{i=1}^{v} \frac{T_i^2}{r_i} - C.F.$	*MST = SST / (v - 1)*	*MST/MSE*
Error	*n – v*	*SSE = by subtraction*	*MSE = SSE / (n - v)*	
Total	*n – 1*	$\sum_{ij} y_{ij}^2 - C.F.$		

The hypothesis that the treatments have equal effects is tested by F-test where F is the ratio *MST / MSE* with *(v - 1)* and *(n - v)* degrees of freedom.

Randomized Complete Block Design

It has been seen that when the experimental units are homogeneous then a CRD should be adopted. In any experiment, however, besides treatments the experimental material is a major source of variability in the data. When experiments require a large number of experimental units, the experimental units may not be homogeneous, and in such situations CRD can not be recommended. When the experimental units are heterogeneous, a part of the variability can be accounted for by grouping the experimental units in such a way that experimental units within each group are as homogeneous as possible. The treatments are then allotted randomly to the experimental units within each group (or blocks). The principle of first forming homogeneous groups of the experimental units and then allotting at random each treatment once in each group is known as local control. This results in an increase in precision of estimates of the treatment contrasts, due to the fact that error variance that is a function of comparisons within blocks, is smaller because of homogeneous blocks. This type of allocation makes it possible to eliminate from error variance a portion of variation attributable to block differences. If, however, variation between the blocks is not significantly large, this type of grouping of the units does not lead to any advantage; rather some degrees of freedom of the error variance is lost without any consequent decrease in the error variance. In such situations it is not desirable to adopt randomized complete block designs in preference to completely randomized designs.

If the number of experimental units within each group is same as the number of treatments and if every treatment appears precisely once in each group then such an arrangement is called a ***randomized complete block design.***

Suppose the experimenter wants to study v treatments. Each of the treatments is replicated r times (the number of blocks) in the design. The total number of experimental units is, therefore, vr. These units are arranged into r groups of size v each. The error control measure in this design consists of making the units in each of these groups homogeneous.

The number of blocks in the design is the same as the number of replications. The v treatments are allotted at random to the v plots in each block. This type of homogeneous grouping of the experimental units and the random allocation of the treatments separately in each block are the two main characteristic features of randomized block designs. The availability of resources and considerations of cost and precision determine actual number of replications in the design.

Analysis

The data collected from experiments with randomized block designs form a two-way classification, that is, classified according to the levels of two factors, *viz.*, blocks and treatments. There are vr cells in the two-way table with one observation in each cell. The

data are orthogonal and therefore the design is called an *orthogonal design.* We take the following model:

$$y_{ij} = \mu + t_i + b_j + e_{ij}\ , \qquad \begin{pmatrix} i = 1,2,...,v; \\ j = 1,2,...,r \end{pmatrix},$$

where y_{ij} denotes the observation from i^{th} treatment in j^{th} block. The fixed effects μ, t_i, b_j denote respectively the general mean, effect of the i^{th} treatment and effect of the j^{th} block. The random variable e_{ij} is the error component associated with y_{ij}. These are assumed to be normally and independently distributed with zero means and a constant variance σ^2.

Following the method of analysis of variance for finding sums of squares due to blocks, treatments and error for the two-way classification, the different sums of squares are obtained as follows: Let $\sum_j y_{ij} = T_i \ \ (i = 1,2,...,v)$ = total of observations from i^{th} treatment and $\sum_j y_{ij} = B_j \quad j = 1,\cdots,r$ = total of observations from j^{th} block. These are the marginal totals of the two-way data table. Let further, $\sum_i T_i = \sum_j B_j = G.$

Correction factor (*C.F.*) = G^2/rv, Sum of squares due to treatments $= \sum_i \frac{T_i^2}{r} - C.F.,$

Sum of squares due to blocks $= \sum_j \frac{B_j^2}{v} - C.F.$, Total sum of squares $= \sum_{ij} y_{ij}^2 - C.F.$

ANALYSIS OF VARIANCE

Sources of variation	**Degrees of freedom (D.F.)**	**Sum of squares (S.S.)**	**Mean squares (M.S.)**	**F**
Blocks	*r - 1*	$SSB = \sum_j \frac{B_j^2}{v} - C.F.$	*MSB = SSB / (r - 1)*	*MSB/MSE*
Treatments	*v - 1*	$SST = \sum_i \frac{T_i^2}{r} - C.F.$	*MST = SST / (v - 1)*	*MST/MSE*
Error	*(r - 1)(v - 1)*	*SSE = by subtraction*	*MSE = SSE / (v - 1)(r - 1)*	
Total	*vr - 1*	$\sum_{ij} y_{ij}^2 - C.F.$		

The hypothesis that the treatments have equal effects is tested by F-test, where F is the ratio *MST / MSE* with *(v - 1)* and *(v - 1)(r - 1)* degrees of freedom. We may then be interested to either compare the treatments in pairs or evaluate special contrasts depending upon the objectives of the experiment. This is done as follows:

The critical difference for testing the significance of the difference of two treatment effects, say $t_i - t_j$ is $C.D. = t_{(v-1)(r-1),\alpha/2}\sqrt{2MSE/r}$, where $t_{(v-1)(r-1),\alpha/2}$ is the value of Student's *t* at the level of significance α and degree of freedom *(v - 1)(r - 1).* If the difference of any two-treatment means is greater than the C.D. value, the corresponding treatment effects are significantly different.

Latin Square Design

Latin square designs are normally used in experiments where it is required to remove the heterogeneity of experimental material in two directions. These designs require that the number of replications equal the number of ***treatments*** or ***varieties***.

Definition 1. A Latin square arrangement is an arrangement of *v* symbols in v^2 cells arranged in *v* rows and *v* columns, such that every symbol occurs precisely once in each row and precisely once in each column. The term *v* is known as the **order** of the Latin square.

If the symbols are taken as *A, B, C, D,* a Latin square arrangement of order 4 is as follows:

A	*B*	*C*	*D*
B	*C*	*D*	*A*
C	*D*	*A*	*B*
D	*A*	*B*	*C*

A Latin square is said to be in the ***standard form*** if the symbols in the first row and first column are in natural order, and it is said to be in the ***semi-standard form*** if the symbols of the first row are in natural order. Some authors denote both of these concepts by the term ***standard form***. However, there is a need to distinguish between these two concepts. The standard form is used for randomizing the Latin-square designs, and the semi-standard form is needed for studying the properties of the orthogonal Latin squares.

Definition 2. If in two Latin squares of the same order, when superimposed on one another, every ordered pair of symbols occurs exactly once, the two Latin squares are said to be **orthogonal**. If the symbols of one Latin square are denoted by Latin letters and the symbols of the other are denoted by Greek letters, the pair of orthogonal Latin squares is also called a **graeco-latin square**.

Definition 3. If in a set of Latin squares every pair is orthogonal, the set is called a set of **mutually orthogonal latin squares (MOLS)**. It is also called a **hypergraeco latin square.**

The following is an example of graeco latin square:

A	B	C	D
B	A	D	C
C	D	A	B
D	C	B	A

α	γ	δ	β
β	δ	γ	α
γ	α	β	δ
δ	β	α	γ

$A\alpha$	$B\gamma$	$C\delta$	$D\beta$
$B\beta$	$A\delta$	$D\gamma$	$C\alpha$
$C\gamma$	$D\alpha$	$A\beta$	$B\delta$
$D\delta$	$C\beta$	$B\alpha$	$A\gamma$

We can verify that in the above arrangement every pair of ordered Latin and Greek symbols occurs exactly once, and hence the two latin squares under consideration constitute a graecolatin square.

It is well known that the maximum number of MOLS possible of order v is $v - 1$. A set of $v - 1$ MOLS is known as a complete set of MOLS. Complete sets of MOLS of order v exist when v is a ***prime or prime power.***

Randomization

According to the definition of a Latin square design, treatments can be allocated to the v^2 experimental units (may be animal or plots) in a number of ways. There are, therefore, a number of Latin squares of a given order. The purpose of randomization is to select one of these squares at random. The following is one of the methods of random selection of Latin squares.

Let a $v \times v$ Latin square arrangement be first written by denoting treatments by Latin letters *A, B, C, etc.* or by numbers *1, 2, 3, etc.* Such arrangements are readily available in the ***Tables for Statisticians and Biometricians*** (Fisher and Yates, 1974). One of these squares of any order can be written systematically as shown below for a 5×5 Latin square:

A	B	C	D	E
B	C	D	E	A
C	D	E	A	B
D	E	A	B	C
E	A	B	C	D

For the purpose of randomization rows and columns of the Latin square are rearranged randomly. There is no randomization possible within the rows and/or columns. For example, the following is a row randomized square of the above 5×5 Latin square;

A	B	C	D	E
B	C	D	E	A
E	A	B	C	D
D	E	A	B	C
C	D	E	A	B

Next, the columns of the above row randomized square have been rearranged randomly to give the following random square:

E	*B*	*C*	*A*	*D*
A	*C*	*D*	*B*	*E*
D	*A*	*B*	*E*	*C*
C	*E*	*A*	*D*	*B*
B	*D*	*E*	*C*	*A*

As a result of row and column randomization, but not the randomization of the individual units, the whole arrangement remains a Latin square.

Analysis of Latin Square Designs

In Latin square designs there are three factors. These are the factors *P*, *Q*, and treatments. The data collected from this design are, therefore, analyzed as a three-way classified data. Actually, there should have been v^3 observations as there are three factors each at *v* levels. But because of the particular allocation of treatments to the cells, there is only one observation per cell instead of *v* in the usual three way classified orthogonal data. As a result we can obtain only the sums of squares due to each of the three factors and error sum of squares. None of the interaction sums of squares of the factors can be obtained. Accordingly, we take the model

$$Y_{ijs} = \mu + r_i + c_j + t_s + e_{ijs}$$

where y_{ijs} denotes the observation in the i^{th} row, j^{th} column and under the s^{th} treatment; $\mu, r_i, c_j, t_s (i, j, s = 1,2,...,v)$ are fixed effects denoting in order the general mean, the row, the column and the treatment effects. The e_{ijs} is the error component, assumed to be independently and normally distributed with zero mean and a constant variance, σ^2.

The analysis is conducted by following a similar procedure as described for the analysis of two-way classified data. The different sums of squares are obtained as below: Let the data be arranged first in a *row* × *column* table such that y_{ij} denotes the observation of (*i*, *j*)th cell of table.

Let $R_i = \sum_j y_{ij} = i^{th}\ row\ total\ (i = 1,2,...,v), C_j = \sum_i y_{ij} = j^{th}\ column\ total\ (j = 1,2,...,v),$

T_s = sum of those observations which come from s^{th} treatment *(s= 1,2,...,v),* $G = \sum_i R_i = grand\ total.$ Correction factor, $C.F. = \frac{G^2}{v^2}$. Treatment sum of squares =

$\sum_s \frac{T_s^2}{v} - C.F.$, Row sum of squares = $\sum_i \frac{R_i^2}{v} - C.F.$, Column sum of squares = $\sum_j \frac{C_j^2}{v} - C.F.$

Analysis of Variance of $v \times v$ Latin Square Design

Sources of Variation	D.F.	S.S.	M.S.	F
Rows	v -1	$\sum_i \frac{R_i^2}{v} - C.F.$		
Columns	v - 1	$\sum_j \frac{C_j^2}{v} - C.F.$		
Treatments	v - 1	$\sum_s \frac{T_s^2}{v} - C.F.$	s_t^2	s_t^2 / s_e^2
Error	$(v - 1)(v - 2)$	By subtraction	s_e^2	
Total	v^2-1	$\sum_{ij} y_{ij}^2 - C.F.$		

The hypothesis of equal treatment effects is tested by *F*-test, where *F* is the ratio of treatment mean squares to error mean squares. If *F* is not significant, treatment effects do not differ significantly among themselves. If *F* is significant, further studies to test the significance of any treatment contrast can be made in exactly the same way as discussed for randomized block designs.

Contrasts Analysis

The main technique adopted for the analysis and interpretation of the data collected from an experiment is the analysis of variance technique that essentially consists of partitioning the total variation in an experiment into components ascribable to different sources of variation due to the controlled factors and error. Analysis of variance clearly indicates a difference among the treatment means. The objective of an experiment is often much more specific than merely determining whether or not all of the treatments give rise to similar responses. For examples, a chemical experiment might be run primarily to determine whether or not the yield of the chemical process increases as the amount of the catalyst is increased. A medical experimenter might be concerned with the efficacy of each of several new drugs as compared to a standard drug. A nutrition experiment may be run to compare high fiber diets with low fiber diets. A plant breeder may be interested in comparing exotic collections with indigenous cultivars. An agronomist may be interested in comparing the effects of biofertilisers and chemical fertilisers. A water technologist may be interested in studying the effect of nitrogen

with Farm Yard Manure over the nitrogen levels without farm yard manure in presence of irrigation.

Contrasts

Let $y_1, y_2, \ldots, y_n$ denote n observations or any other quantities. The linear function $C = \sum_{i=1}^{n} l_i y_i$, where l_i's are given number such that $\sum_{i=1}^{n} l_i = 0$, is called a *contrast* of $y_i's$.

Let $y_1, y_2, \ldots, y_n$ be independent random variables with a common mean μ and variance σ^2. The expected value of the random variable C is zero and its variance is $\sigma^2 \sum_{i-1}^{n} l_i^2$. In what follows we shall not distinguish between a contrast and its corresponding random variable.

Sum of squares (s.s.) of contrasts. The sum of squares due to the contrast C is defined as $C^2 / \sigma^{-2} Var(C) = C^2 / \left(\sum_{i=1}^{n} l_i^2 \right)$. Here σ^2 is unknown and is replaced by its unbiased estimate, *i.e. mean square error*. It is known that this square has a $\sigma^2 \chi^2$ distribution with one degree of freedom when the $y_i's$ are normally distributed. Thus the sum of squares due to two or more contrasts has also a $\sigma^2 \chi^2$ distribution if the contrasts are independent. Multiplication of any contrast by a constant does not change the contrast. The sum of squares due to a contrast as defined above is not evidently changed by such multiplication.

Orthogonal contrasts. Two contrasts, $C_1 = \sum_{i=1}^{n} l_i y_i$ and $C_2 = \sum_{i=1}^{n} l_i y_i$ are said to be orthogonal if and only if $\sum_{i=1}^{n} l_i m_i = 0$. This condition ensures that the covariance between C_1 and C_2 is zero.

When there are more than two contrasts, they are said to be mutually orthogonal if they are orthogonal pair wise. For example, with four observations y_1, y_2, y_3, y_4, we may write the following three mutually orthogonal contrasts:

(i) $y_1 + y_2 - y_3 - y_4$

(ii) $y_1 - y_2 - y_3 + y_4$

(iii) $y_1 - y_2 + y_3 - y_4$

The sum of squares due to a set of mutually orthogonal contrasts has a $\sigma^2 \chi^2$ distribution with as many degrees of freedom as the number of contrasts in the set.

Multiple Comparison Procedures

Duncan's Multiple Range Test

A widely used procedure for comparing all pairs of means is the multiple range test developed by Duncan (1955). The application of Duncan's multiple range test (*DMRT*) is similar to that of *lsd* test. *DMRT* involves the computation of numerical boundaries that allow for the classification of the difference between any two treatment means as significant or non-significant. *DMRT* requires computation of a series of values each corresponding to a specific set of pair comparisons unlike a single value for all pairwise comparisons in case of *lsd.* It primarily depends on the standard error of the mean difference as in case of *lsd.* This can easily be worked out using the estimate of variance of an estimated elementary treatment contrast through the design.

For application of the *DMRT* rank all the treatment means in decreasing or increasing order based on the preference of the character under study.

Tukey Method for All Pairwise Comparisons

Tukey (1953) proposed a method for making all possible pairwise treatment comparisons. The test compares the difference between each pair of treatment effects with appropriate adjustment for multiple testing. This test is also known as Tukey's honestly significant difference test or Tukey's HSD. It may be mentioned here that Tukey's method is the best for all pairwise treatment comparisons. It can be used for completely randomized designs, randomized complete block designs and balanced incomplete block designs. It is believed to be applicable (conservative, true α level lower than stated) for other incomplete block designs as well, but this has not yet been proven. It can be extended to include all contrasts but Scheffe's method is generally better for these types of contrasts.

Dunnett Method for Treatment-Versus-Control Comparisons

Dunnett (1955) developed a method of multiple comparisons for obtaining a set of simultaneous confidence intervals for preplanned treatment-versus-control contrasts $t_i - t_1 (i = 2,\ldots,v)$ where level 1 corresponds to the control treatment. The intervals are shorter than those given by the Scheffe, Tukey and Bonferroni methods, but the method should not be used for any other type of contrasts. For details on this method, a reference may be made to Dunnett (1955, 1964) and Hochberg and Tamhane (1987). In general this procedure is, therefore, best for all treatment-versus-control comparisons. It can be used for completely randomized designs, randomized complete block designs. It can also be used for balanced incomplete block designs but not in other incomplete block designs without modifications to the corresponding multivariate t-distribution tables given in Hochberg and Tamhane (1987).

References

Kempthorne, O. (1977). Why randomize? *Journal of Statistical Planning and Inference*, **1**, 1-25.

Dean, A. and Voss, D. (1999). *Design and Analysis of Experiments*. Springer Text in Statistics, New York.

Fisher, R.A. and Yates, F. (1963). *Statistical Tables For Biological, Agricultural and Medical Research*. Longman Group Ltd., England.

Parsad, Rajender and Gupta, V.K. Basic Experimental Designs. E book chapter available at http://www.iasri.res.in/ebook/EB_SMAR/e-book_pdf%20files/Manual%20III/2-Basic%20Experiments.pdf

Stability Analysis - Use of Additive Main Effects and Multiplicative Interaction (AMMI) Model in Crop Improvement

1. Introduction

United Nations projections estimate that the world population will continue to grow from the current 6 billion to about 10 billion by 2050 [FAO, 1996]. The increase in population and the subsequent rise in the demand for agricultural produce are expected to be greater in regions where production is already insufficient, in particular in Sub-Saharan Africa and South Asia. The necessary increase in agricultural production represents a huge challenge to local farming systems and must come mainly from increased yield per unit area, given the limited scope for extension of cultivated land worldwide. To meet this requirement various crop improvement programmes all over the world have been initiated.

Under any crop improvement programme a sample of promising genotypes are performance tested each year at a number of sites, representing the major growing area of the crop with a view to identify genotypes which possess the dual qualities of high-yield sustainability and low sensitivity to adverse changes in environmental condition. One of the important steps here is to assess the performance of improved genotypes in multi-environment (multi-location, multi-year or both) trials. Quite often it is observed that varieties perform differently in different environments. A specified difference in environment may produce differential (different) effect on genotype. This interplay of genetic and non-genetic effects causing differential relative performances of genotypes in different environments is called Genotype x Environment Interaction (GEI). Presence of GEI causes difficulty in identifying superior genotypes. Not withstanding its importance GEI is often a distraction in genetical analysis for which effort is usually made to overcome such interactions. One way of reducing GEI is through resistant breeding, usually adopted by plant breeders. Since only a minor part of the GEI can be attributed to controllable environmental determinants, much reduction in interaction can not be achieved. The most practical alternative is to produce progressively better adapted populations to the existing environments.

A detailed description and discussion of various aspects of GE interaction analysis is available in numerous review articles (Freeman, 1973; Hill, 1975; Denis and Vincourt, 1982; Westcott, 1986; Lin *et al*., 1986; Becker and Léon, 1988; Crossa, 1990; Romagosa and Fox, 1993; Cooper and DeLacy, 1994; van Eeuwijk, 1995; Brancourt-Hulmel *et al*., 1997; Kang, 1998), in papers included in the books edited by Williams (1976), Kang (1990), Kang and Gauch (1996), Cooper and Hammer (1996) and Kang (2002), and in the monographs by Gauch (1992), Prabhakaran and Jain (1994) and Basford and Tukey (2000). In this lecture, analysis of GE data through AMMI approach will be mainly discussed along with the examples.

2. AMMI Model

Gauch (1988, 1992) has advocated the use of AMMI analysis for yield trials. Gauch and Zobel (1988) compared the performance of AMMI analysis with the ANOVA approach and regression approach and found that ANOVA fails to detect a significant interaction component and the regression approach accounts only a small portion of the interaction sum of squares only when the pattern fits a specific regression model.

The AMMI model for T genotypes and S environments is given as

$$Y_{ij} = \mu + g_i + e_j + \sum_{n=1}^{n'} \lambda_n \alpha_{in} \gamma_{jn} + \theta_{ij} \qquad (1)$$

$$\theta_{ij} \sim N(0, \sigma^2); \ i = 1,2,..., T; \ j = 1,2,..., S.$$

where, Y_{ij} is the mean yield of i*th* genotype in the j*th* environment; μ is the general mean; g_i is the i*th* genotypic effect; e_j is the j*th* location effect; λ_n is the eigen value of the PCA axis *n*; α_{in} and γ_{jn} are the i*th* genotype j*th* environment PCA scores for the PCA axis *n*; θ_{ij} is the residual; n' is the number of PCA axes retained in the model. Ordinarily the number n' is judged on the basis of empirical consideration of F-test of significance [Gauch (1988, 1992)]). The residual combines the PCA scores from the *N - n'* discarded axes, where *N* = min (t-1, s-1). The other constraints in the model (1) are $\sum_i \alpha_{in}^2 = \sum_j \gamma_{jn}^2 = 1 \ \forall$ n; $\sum_i \alpha_{in}\alpha_{in*} = \sum_j \gamma_{jn}\gamma_{jn*} = 0$, $n \neq n^*$; and $\lambda_1 > \lambda_2 > \cdots > \lambda_{n'} > 0$. The model in (1) can be reparameterized as

$$Y_{ij} = \mu + g_i + e_j + Z_{ij} \qquad (2)$$

where $Z_{ij} = \sum_{n=1}^{n'} \lambda_n \alpha_{in} \gamma_{jn} + \theta_{ij}$.

Let the estimate of interaction in the (i, j) th cell Z_{ij} be $\hat{Z}_{ij} = Y_{ij} - \hat{\mu} - \hat{g}_i - \hat{e}_j$. Using matrix notation, denote **Z** = ($\hat{Z}_{ij}$) a matrix of order T x S. Now, the estimates of the parameters of the model are:

$\hat{\lambda}_n$ = the non-zero eigen values of **Z′Z** (in descending order), and

$\hat{\alpha}_{in}$ = the principal components of the row sum of squares and cross product matrix **ZZ′**

$\hat{\gamma}_{jn}$ = the principal components of the column sum of squares and cross product matrix **Z′Z**

Using these we can write

$$\hat{z}_{ij} = \sum_{n=1}^{N} \hat{\lambda}_n \hat{\alpha}_{in} \hat{\gamma}_{jn} \qquad (3)$$

It follows that, $\alpha^*_{in} = \lambda_n^c \hat{\alpha}_{in}$ is the ith genotype PCA score for the *n*th axis, and $\gamma^*_{jn} = \lambda_n^{1-c} \hat{\gamma}_{jn}$ is the nth PCA score of the jth environment; where *c* is a scaling constant varying between 0 to 1.

Also, Using factor analytic decomposition, **Z** may be written as

$$\mathbf{Z} = \mathbf{ADB'} \qquad (4)$$

where **A** is TxN orthonormal matrix, **D** is NxN diagonal matrix with elements $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_{n'} \geq \ldots \lambda_N$, **B** is NxS orthonormal matrix, N is the rank of **Z**. The matrices **A**, **D** and **B** of equation (4) can be obtained from the eigen vectors and eigen values of **ZZ'** of the order TxT. The matrix **A** consists of the eigen vectors (principal components α_{in}) of **ZZ'** and the diagonal matrix **D** with square root of eigen values as diagonal elements of **ZZ'**. The matrix **B** consists of the eigen vectors (principal components γ_{jn}) can be obtained by solving $\mathbf{B} = \mathbf{Z'AD^{-1}}$. For many practical situations, the number of PCA axes to be retained is determined by testing the mean square of each axis with the estimate of residual through F-statistic [Gollob (1968), Gauch (1988)]. The mean sum of squares of each PCA axis is equal to the ratio of square of the corresponding eigen value and the degree of freedom of each axis obtained as T + S –1 –2*n*.

Biplots

The model formulation for AMMI shows its interaction part consists of summed orthogonal products. Because of this form the interaction lends itself to graphical display in the form of so-called biplots (Gabriel, 1971). Let us start with AMMI and assume that either two terms suffice for an adequate description of the interaction. For AMMI the interaction consists then of the sum of two products: $\alpha^*_{i1}\gamma^*_{j1} + \alpha^*_{i2}\gamma^*_{j2}$. The choice of the scaling constant *c* depends on the purposes of the analysis. Usually one is more interested in the genotypes and *c* is chosen equal to one (Kempton [12]). The features of the biplots, however, are not too critically dependent on *c*, and *c* = 0.5 may suit well for most problems. The genotypic scores, α^*_{i1} and α^*_{i2}, are now interpreted as coordinates for a planar depiction of the genotypes, and the environmental scores, γ^*_{j1} and γ^*_{j2}, for a similar depiction of the environments. The scores determine the end points of genotypic and environmental vectors, which depart from the origin. Simple geometry reveals that the interaction between a genotype *i* and an environment *j* can be obtained from a projection of either vector onto the other. The reason is that the interaction according to an AMMI model with two product terms of interaction, $\alpha^*_{i1}\gamma^*_{j1} + \alpha^*_{i2}\gamma^*_{j2}$, is equal to the inner product between vectors $(\alpha^*_{i1}, \alpha^*_{i2})$ and $(\gamma^*_{j1}, \gamma^*_{j2})$, or the projection of either vector onto the other, times the length of the vector on which projection takes place. It is easy to read from a biplot the relative interactions that genotypes exhibit in a particular environment.

Example

Shown below is the pod yield of 15 varieties of ground nut crop raised at 20 locations. The experimental design used is RCBD at each locations with 3 replications.

	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12	L13	L14	L15	L16	L17	L18	L19	L20
G1	1773	880	2841	2020	856	1382	1458	282	1190	1001	2708	1832	1188	2252	1583	2014	2199	810	1033	992
G2	1715	861	2497	2020	505	1104	1153	275	1394	882	1956	1907	729	1658	1285	1986	2014	865	600	842
G3	1241	424	3266	1717	1148	1225	1130	113	701	705	1688	1568	1153	2073	1303	2361	2893	1028	1000	997
G4	1472	917	3172	2222	1505	1475	1222	632	1308	334	2833	1157	792	956	1374	2570	611	486	333	1049
G5	1208	1435	3625	1919	903	1432	921	862	1081	539	2303	1778	577	1132	1368	2691	495	639	300	877
G6	1893	1310	2716	2374	1320	1476	1482	680	1468	591	2877	2333	1005	2636	1438	2812	1968	963	1100	1413
G7	1852	1169	2527	2222	903	1220	1407	455	1637	521	2042	1732	1285	2046	1333	2500	2060	949	633	877
G8	1266	993	2245	1869	292	972	1171	275	1419	767	2184	2037	799	1749	1368	2083	1537	732	667	965
G9	1736	792	2376	2172	981	1113	1051	364	1579	364	2940	1500	819	1668	1041	1944	2431	1000	633	967
G10	1442	695	2800	2071	1051	1890	1051	605	1684	67	2083	1419	1146	1295	1750	2726	1713	50	600	1166
G11	1530	1055	2643	2172	1412	1049	1051	567	1211	174	1977	1963	1083	2063	1319	1789	1435	944	633	1309
G12	1697	1222	2770	2273	1759	1343	1153	572	1169	353	2014	2222	792	1634	1319	2271	2014	1176	1200	1026
G13	1637	1097	2715	2071	1806	1158	1199	636	1269	437	1574	1843	958	1719	1299	2014	2431	1014	1033	1379
G14	1641	1403	2712	2071	792	1037	1199	757	1296	643	2347	1889	1035	1551	1375	1993	2222	875	933	1092
G15	1727	1139	2452	2071	481	883	1519	299	1330	366	1535	1574	1070	1940	1146	1514	2208	745	567	904

Using the SAS programme, SISGYS2 (Rao *et al.*, 2004), the ANOVA for the ground nut data is obtained and presented in Table 1. It is evident from Table 1 that the use of biplots to explain efficiently the interaction is very much limited, since the first two PCA axes explain only 55% of the total interaction variation. Hence at least six axes must be retained to explain GEI. All other calculations can be seen from the out put.

Table 1. AMMI analysis of variance for groundnut data

Source	Df	Sum of squares	Mean square	Variance ratio
Genotypes	14	3565604.00	254686.00	12.63**
Environments	19	107622796.00	5664357.70	280.86**
G x E interaction	266	25408293.00	95519.90	4.74**
PCA1	32	10240492.00	320015.38	12.05**
PCA2	30	3899700.40	129990.01	4.90**
PCA3	28	2795398.80	99835.67	3.76**
PCA4	26	2377000.80	91423.11	3.44**
PCA5	24	1961588.90	81732.87	3.08**
PCA6	22	1372729.60	62396.80	2.35**
Remainder	104	2761382.50	26551.75	
Average error	560	11294080.00	20168.00	

3. Simultaneous selection of varieties for yield and stability

Genotype x environment interaction continues to be a challenging issue among plant breeders, geneticists, and production agronomists who conduct crop performance trials across diverse environments. GEI can reduce progress from selection. The methods of partitioning GEI into components attributable to each genotype measure the contribution of each genotype to GEI. A universally acceptable selection criterion that takes GEI into consideration does not exist. Whenever an interaction is significant, the use of main effects, e.g. overall genotypes means across environments, is questionable. Stability of performance should be considered an important aspect of yield trials. Researchers need a statistic that provides a reliable measure of stability or consistency of performance across a range of environments, particularly one that reflects the contribution of each genotype to the total GEI. However, the stability measure alone is of limited use. To be of practical utility in a breeding or cultivar testing programme, both stability and yield (or any other trait) must be considered simultaneously so as to make selection of genotypes more precise and reliable. Also integration of stability of performance with yield through suitable measures will reduce the effect of GEI and will help in selecting cultivars in a more refined manner.

Several methods of simultaneous selection for yield and stability and relationships among them were discussed by Kang and Pham (1991). Kang (1993) discussed the reasons for emphasizing stability in the selection process. The development and use of Yield-Stability statistic (YSi) has enabled incorporation of stability in the selection process (Kang, 1993). A computer program (STABLE) for calculating this statistic is available free of charge (Kang and Magari, 1995). Kang's Yield-Stability statistic (Kang, 1993) has been evaluated and found to be useful for recommending varieties (Pazdernik et al., 1997; Hussein et al., 2000). However, Bajpai and Prabhakaran (2000) observed that Kang's rank-sum method has an inherent weakness that it is weighing heavily towards yield performance, apart from the arbitrariness in the scoring procedure. Accordingly they proposed three new indices (I1, I2, I3), which were found to be superior to Kang (1993) indices.

AMMI based Selection indices for cultivar x environment data

Rao *et al.* (2004) proposed a new stability measure and incorporating it as a stability component, a new family of selection indices is constructed. As evident from literature on AMMI the scope of biplots is very much limited. The inferences drawn from biplots will be valid only when the first principal component axis (PCA) or the first two PCAs explain maximum interaction variation. Whenever more than two axes are retained in the AMMI model, the biplot formulation of interaction will fail. Consequently the conclusions drawn on stability of varieties will be imprecise. However, the plant breeders would like to identify varieties which are stable and high yielding when the PCA axes retained in the AMMI model will be more than two, if the axes together accumulate considerable portion of interaction variation. Suppose that n' of the N axes are retained in the AMMI model to explain GEI, then the stability measure of i-th variety can now be determined as the end point of its vector α^*_{1i}, α^*_{2i}, . . . , $\alpha^*_{n'i}$ from the origin $0'_{n'x1}$. This can also be taken as the squared Euclidean distance between the vector $\gamma = (\alpha^*_{1i}, \alpha^*_{2i}, \ldots, \alpha^*_{n'i})'$ from the origin, in the n'- dimensional Euclidean space.

$$\text{ASTAB}_i = d_i(\gamma, 0) = \alpha_{1i}^{2*} + \alpha_{2i}^{2*} + \ldots + \alpha_{n'i}^{2*} = \sum_{n=1}^{n'} \alpha_{ni}^{2*} = \sum_{n=1}^{n'} \lambda_n \alpha_{ni}^2 \qquad (5)$$

The algebric expression of the above said stability measure can also be derived from the spetral decomposition of the **ZZ'** matrix. As we know that

$$\mathbf{ZZ}' = \lambda_1 \alpha_1 \alpha_1' + \lambda_2 \alpha_2 \alpha_2' + \cdots + \lambda_n \alpha_n \alpha_n' + \cdots + \lambda_N \alpha_N \alpha_N' ,$$

the ith diagonal element of **ZZ'**, i.e., $\sum_{j=1}^{s} Z_{ij}^2$, is nothing but the interaction effects of i-th genotype over s environments. Therefore

$$\sum_{j=1}^{s} Z_{ij}^2 = \lambda_1 \alpha_{1i}^2 + \lambda_2 \alpha_{2i}^2 + \cdots + \lambda_N \alpha_{Ni}^2 = \sum_{n=1}^{N} \lambda_n \alpha_{ni}^2 \qquad (6)$$

The proposed stability measure of the i-th genotype in (5), mentioned earlier as a squared Euclidean distance, will be equal to the expression given in (6) when N = n', n' being the number of PCA axes retained in the AMMI model to explain the larger part of the GEI variation. A variety is considered as highly stable when the value of $ASTAB_i$ is small or closer to zero. The stability measure given in (5) will now be used as a stability component in the simultaneous selection index. A new family of simultaneous selection indices can thus be evolved, which consists of a yield component, measured as the ratio of the average performance of the ith genotype to the overall mean performance of the genotypes under test and a stability component, measured as the ratio of stability information ($1/ASTAB_i$) of the *i* th genotype to the mean stability information of the genotypes under test. The expression of the index is given as

$$I_2 = \frac{\bar{Y}_{i.}}{\bar{\bar{Y}}_{..}} + \psi \frac{(1/ASTAB_i)}{(\frac{1}{t}\sum_{i}^{t}\frac{1}{ASTAB_i})} \qquad (7)$$

where ψ is the ratio of the weights given to the stability component (w_2) and yield component (w_1) with a restriction that $w_1 + w_2 = 1$. The family of indices will consists of four indices I_{21}, I_{22}, I_{23} and I_{24} by considering the value of ψ as 1.0, 0.66, 0.43 and 0.25 respectively. The performance of the new family is assessed by standard techniques like, the percentage of high yielders and highly stable varieties present in the top 50% of the varieties selected based on the indices. The rank correlations are worked out between yield based ranks and index based ranks, stability based ranks and index based ranks. It is evident from Table 1that at least six axes must be retained for using the proposed simultaneous selection indices. The rank orders based on yield, stability ($ASTAB_i$), proposed index and Bajpai index for each genotype and for different ψ values are presented in Table 2. Table 3 shows the rank correlations between yield, stability with the proposed indices and Bajpai indices. Significant correlations of order 0.59, 0.61, 0.64 and 0.78 are observed between yield and proposed index when value of α is taken as 1.0, 0.67, 0.43 and 0.25 respectively, whereas with the Bajpai's index the correlations are to the extent 0.49, 0.51, 0.55 and 0.64. Further, the correlations indicate superiority of the proposed index over the Bajpai's index. Also Table 3 indicates the extent of high linear relationship between the rank orders of proposed index with the stability. Besides, these correlations are at par with the correlations observed between stability and Bajpai's index. The proportion of high yielders and stable performers present in the 50% top selected genotypes based on simultaneous selection index values are presented in Table 4. From this table, it is evident that among the top 50% varieties selected based on the proposed indices, around 70% are the high yielders and 85% are high stable performers. Since the proposed indices show significant correlations with both high yield and stability as

well as selects large proportion of high yielders and stable performers, they can be safely recommended to the breeders and production agronomists.

A computer programme

A SAS programme named SISGYS2 is developed for selecting genotypes simultaneously for yield and stability. This programme requires genotype means over replications from individual locations. The input file should be in Excel and should contain a single field with *yld* as first row and the subsequent rows should be the mean yield over replications for each genotype nested within locations. The input file should be named 'data.xls'. The number of genotypes and the number of locations should be provided inside the programme codes. The programme calculates the following steps: (i) genotype's mean performance (ii) genotype's stability measure ($ASTAB_i$) or d_i (iii) genotype's index value I. Based on the index values genotypes are ranked. The genotype with highest index value will be ranked 1. The SAS code developed for the purpose is given in ANNEXURE-I. To demonstrate the programme, the groundnut yields of 15 varieties in 20 locations, under cultivar X location set up, are taken. The input data is arranged in a nested fashion as genotypes within locations and output (result) is as below:

INDEX VALUE	RANK	YIELD (t/ha)	RANK1	STABILITY ($x\ 10^6$)	RANK2
1.33	5	1.51	2	1.98	8
1.22	8	1.31	12	1.76	6
1.11	12	1.39	7	4.04	13
1.03	14	1.32	11	5.95	14
1.01	15	1.30	13	6.03	15
1.59	1	1.69	1	1.31	3
1.46	3	1.47	4	1.23	2
1.19	10	1.27	15	1.79	7
1.21	9	1.37	8	2.18	9
1.13	11	1.37	10	3.24	12
1.33	6	1.37	9	1.41	4
1.36	4	1.50	3	1.71	5
1.23	7	1.46	5	2.67	10
1.47	2	1.44	6	1.15	1
1.07	13	1.27	14	3.16	11

Table 2. Effect of variation of weights on the rank orders of groundnut varieties in the simultaneous selection indices

Variety	Yield (tones / hectare)	Yield based Rank	Stability based Rank	ψ = 1.0		ψ = 0.67		ψ = 0.43		ψ = 0.25	
				Proposed index based rank	Bajpai's index based rank	Proposed index based rank	Bajpai's index based rank	Proposed index based rank	Bajpai's index based rank	Proposed index based rank	Bajpai's index based rank
G1	1.51	2	8	6	5	6	5	6	5	5	4
G2	1.34	12	6	7	3	7	3	7	4	8	5
G3	1.38	7	13	13	13	13	13	12	13	12	13
G4	1.32	11	14	14	14	14	14	14	14	14	14
G5	1.30	13	15	15	15	15	15	15	15	15	15
G6	1.69	1	3	2	4	1	4	1	3	1	3
G7	1.46	4	2	3	2	3	2	3	2	3	2
G8	1.27	15	7	8	8	8	8	8	8	10	9
G9	1.37	8	9	9	10	9	10	9	10	9	10
G10	1.36	10	12	11	12	11	12	11	12	11	11
G11	1.37	9	4	4	6	4	7	4	7	6	7
G12	1.50	3	5	5	7	5	6	5	6	4	6
G13	1.46	5	10	10	9	10	9	10	9	7	8
G14	1.44	6	1	1	1	2	1	2	1	2	1
G15	1.27	14	11	12	11	12	11	13	11	13	12

Table 3. Rank correlations between simultaneous selection indices and yield, stability for groundnut data.

Index Type	**Weightage on components of index**							
	ψ = 1.00		**ψ = 0.67**		**ψ = 0.43**		**ψ = 0.25**	
	Yield	**Stability**	**Yield**	**Stability**	**Yield**	**Stability**	**Yield**	**Stability**
Proposed index	0.596*	0.982**	0.614**	0.975**	0.639**	0.968**	0.782**	0.914**
Bajpai's index	0.493^{NS}	0.946**	0.514*	0.943**	0.553*	0.953**	0.639**	0.932**

Table 4. Proportion of high yielders (HY) and highly stable performers (HSP) present out of top 50% genotypes selected on the basis of simultaneous selection indices

Index Type	**Weightage (α)**							
	ψ = 1.00		**ψ = 0.67**		**ψ = 0.43**		**ψ = 0.25**	
	HY	**HSP**	**HY**	**HSP**	**HY**	**HSP**	**HY**	**HSP**
Proposed index	0.71 (6,14,7,12,1)	0.86 (6,14,7,11,12,2)	0.71 (6,14,7,12,1)	0.86 (6,14,7,11,12,2)	0.71 (6,14,7,12,1)	0.86 (6,14,7,11,12,2)	0.86 (6,14,7,12,1,13)	0.86 (6,14,7,11,12,2)
Bajpai's index	0.71 (14,7,6,1,12)	0.86 (14,7,6,2,12,11)	0.71 (14,7,6,1,12)	0.86 (14,7,6,2,12,11)	0.71 (14,7,6,1,12)	0.86 (14,7,6,2,12,11)	0.71 (14,7,6,1,12)	0.86 (14,7,6,2,12,11)

References

Bajpai, P.K. and Prabhakaran, V.T. 2000. A new procedure of simultaneous selection for high yielding and stable crop genotypes. *Ind. J. Genet.*, 60(2), 141-146.

Basford, K.E. and Tukey, J.W. 2000. *Graphical analysis of multiresponse data: illustrated with a plant breeding trial*. Boca Raton, FL, Chapman & Hall/CRC Press.

Becker, H.C. and Léon, J. 1988. Stability analysis in plant breeding. *Plant Breed*., 101: 1-23.

Brancourt-Hulmel, M., Biarnès-Dumoulin, V. and Denis, J.B. 1997. Points de repère dans l'analyse de la stabilité et de l'interaction génotype-milieu en amélioration des plants. *Agronomie*, 17: 219-246.

Cooper, M. & DeLacy, I.H. 1994. Relationships among analytical methods used to study genotypic variation and genotype-by-environment interaction in plant breeding multi-environment experiments. *Theor. Appl. Genet*., 88: 561-572.

Cooper, M. and Hammer, G.L. (eds). 1996. *Plant adaptation and crop improvement*. Wallingford, UK, CABI.

Crossa, J., Gauch, H.G. and Zobel, R.W. 1990. Additive main effects and multiplicative interaction analysis of two international maize cultivar trials. *Crop Sci*., 30: 493-500.

Denis, J.B. and Vincourt, P. 1982. Panorama des méthodes statistiques d'analyse des interactions génotype x milieu. *Agronomie*, 2: 219-230.

FAO**.** 1996. *Food requirements and population growth*. Technical Background Document No. 4. Rome.

Freeman, G.H. 1973. Statistical methods for the analysis of genotype-environment interaction. *Heredity*, 31: 339-354.

Gabriel, K.R. 1971. The biplot-graphical display of matrices with applications to principal component analysis. *Biometrika*, 58, 453-467.

Gauch, H.G. 1988. Model selection and validation for yield trials with interaction. *Biometrics*, 44: 705-715.

Gauch, H.G. 1992. *Statistical analysis of regional yield trials: AMMI analysis of factorial designs*. Amsterdam, Elsevier.

Gauch, H.G.and Zobel, R.W. 1988. Predictive and postdictive success of statistical analysis of yield trial. *Theor. Appl. Genet.* , 76, 1-10.

Gollob, H.F. (1968). A statistical model which combines features of factor analytic and analysis of variance techniques. *Psychometrika*, **33**, 73-115.

Hill, J. 1975. Genotype-environment interactions - A challenge for plant breeding. *J. Agric. Sci., Camb*., 85: 477-493.

Hussein, M.A., A. Bjornstad, and A.H. Aastveit. 2000. SASG x ESTAB: A SAS program for computing genotype x environment stability statistics. *Agron. J*. 92:454-459.

Kang, M. S. 1993. Simultaneous selection for yield and stability in crop performance trials: Consequences for growers. *Agron. J*., 85, 754-757.

Kang, M.S. and Gauch, H.G. (eds). 1996. *Genotype-by-environment interaction*. Boca Raton, FL, CRC Press.

Kang, M.S. (ed.) 1990. *Genotype-by-environment interaction and plant breeding*. Baton Rouge, LA, Louisiana State Univ.

Kang, M.S. (ed.) 2002. *Quantitative genetics, genomics, and plant breeding*. Wallingford, UK, CABI.

Kang, M.S. 1998. Using genotype-by-environment interaction for crop cultivar development. *Adv. Agron*., 62: 199-252.

Kang, M.S., and Magari, R. 1995. STABLE: Basic program for calculating yield-stability statistic. *Agron. J*., 87, 276-277.

Kang, M.S., and Pham, H.N. 1991. Simultaneous selection for high yielding and stable crop genotypes. *Agron. J*., 83,161-165.

Pazdernik, D.L., Hardman, L.L. and Orf, J.H. 1997. Agronomic performance of soybean varieties grown in three maturity zones of Minnesota. *J. Prod. Agric*., 10, 425-430.

Prabhakaran, V.T. and Jain, J.P. 1994. *Statistical techniques for studying genotype-environment interactions*. New Delhi, South Asian Publ.

Rao, A.R., Prabhkaran, V.T. and Singh, A.K. 2004. Development of statistical procedures for selecting genotypes simultaneously for yield and stability. *IASRI Publication*.

Romagosa, I., Fox, P.N., Garcìa del Moral, L.F., Ramos, J.M., Garcìa del Moral, B., Roca de Togores, F. & Molina-Cano, J.L. 1993. Integration of statistical and physiological analyses of adaptation of near-isogenic barley lines. *Theor. Appl. Genet*., 86: 822-826.

van Eeuwijk, F.A. 1995. Linear and bilinear models for the analysis of multi-environment trials. I. An inventory of models. *Euphytica*, 84: 1-7.

Westcott, B. 1986. Some methods of analysing genotype-environment interaction. *Heredity*, 56: 243-253.

Williams, W.T. (ed.) 1976. *Pattern analysis in agricultural science*. Amsterdam, Elsevier.