



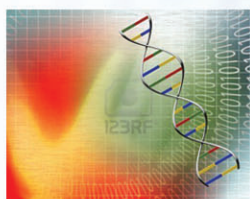
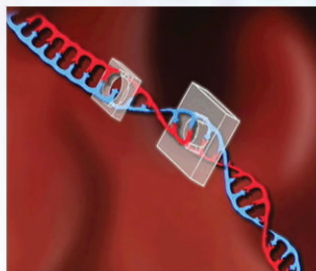
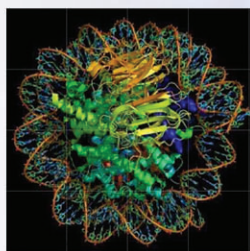
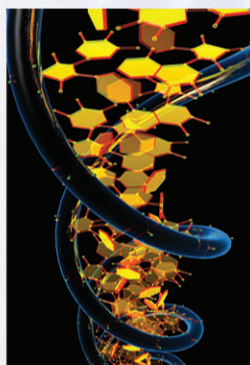
Training Manual



DBT Sponsored
Training Programme

Whole genome sequencing and annotation

February 21st – 25th, 2012



INDIAN INSTITUTE OF SPICES RESEARCH

(Indian Council of Agricultural Research)

Kozhikode - 673012, Kerala





**DBT Sponsored
Training Program on
Whole genome sequencing and annotation
21 – 25 February 2012**

TRAINING MANUAL



**INDIAN INSTITUTE OF SPICES RESEARCH
(INDIAN COUNCIL OF AGRICULTURAL RESEARCH)
CALICUT – 673 012, KERALA**

Published by

M. Anandaraj
Director

Organized by

Santhosh J Eapen, Co-ordinator

Compiled & Edited by

Santhosh J Eapen
Rosana D B
Arun John
A. Chandrasekar

The manual is an in-house publication intended for training purposes only and is not for public circulation.

Copyright © 2012 IISR. All rights reserved. Reproduction and redistribution prohibited without approval.

CONTENTS

Sl.No	Topics	Page No
1	Whole Genome Sequencing	1
2	Data Formats & Quality Check Analysis	15
3	Genome Alignment & Assembly	40
4	Genome Annotation	48
5	Whole Genome Re-sequencing & NGS Data Analysis	53
6	Transcriptome sequencing – a case study in <i>Piper</i>	62
7	Comparative Genomics	70
8	APPENDIX	81

1. Whole Genome Sequencing

Whole genome sequencing or complete genome sequencing is a laboratory process that determines the complete DNA sequence of an organism's genome at a single time. The information garnered from sequencing will provide the raw data for the exploding field of bioinformatics, where computer science and biology live in symbiotic harmony to derive meaningful knowledge from DNA sequences, which can define various problems of biological research.

The information that can be gathered from whole genome sequencing is as follows.

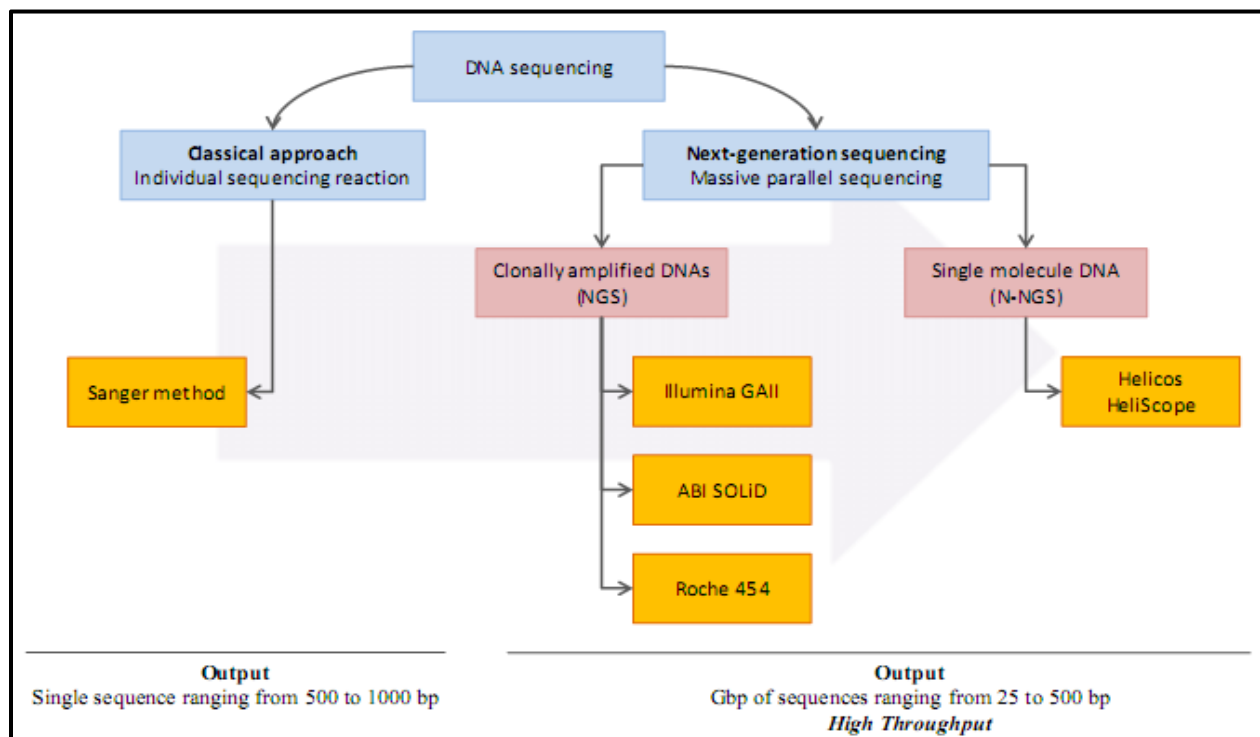
- Gene number, exact locations, and functions
- Gene regulation or promoter region
- DNA sequence organization
- Chromosomal structure and organization
- Noncoding DNA types, amount, distribution, information content, and functions
- Predicted Vs experimentally determined gene function
- Evolutionary conservation among organisms
- Protein conservation (structure and function)
- Proteomes (total protein content and function) in organisms
- Correlation of SNPs (single-base DNA variations among individuals) with health and disease
- Disease-susceptibility prediction based on gene sequence variation
- Genes involved in complex traits and multigene diseases

Whole genome shotgun sequencing for small (4000 to 7000 basepair) genomes was already in use in 1979; this include broader application benefited from pairwise end sequencing, known colloquially as double-barrel shotgun sequencing [1]. In 1995 Roach *et al.* [2] introduced the innovation of using fragments of varying sizes, and demonstrated that a pure pairwise end-sequencing strategy would be possible on large targets. The strategy of Roach *et al.* was subsequently adopted by The Institute for Genomic Research (**TIGR**) to sequence the genome of the bacterium *Haemophilus influenza* in 1995 [3]. Further **Celera Genomics** to sequence the *Drosophilla melanogaster* (fruit fly) genome in 2000 [4]. Subsequently the whole-genome shotgun sequencing was believed to be limited by both the sheer size of large genomes and by the complexity added by the high percentage of repetitive DNA (greater than 50% for the human genome) present in large genomes. It was not widely accepted that a full-genome shotgun sequence of a large genome would provide reliable data. For these reasons, other strategies called **hierarchical sequencing** (2000- 2005) that lowered the computational load of sequence assembly has been used to complete the Human Genome Project [5, 6], most of the human genome was sequenced at 12X or greater *coverage*. By 2004 / 2005, pyrosequencing [7] had been brought to commercial viability by 454 Life Sciences. This new sequencing methods

generated reads much shorter than from Sanger sequencing: initially about 100 bases, now 400-500 bases. However, due to the much higher throughput and lower cost than Sanger sequencing, the adoption of this technology by genome centers pushed development of sequence assemblers to deal with this new type of sequences. The genome of *Mycoplasma genitalium* genome has been completed using pyrosequencing).

Although shotgun sequencing and hierarchical sequencing was the most advanced technique for sequencing genomes from about 1995–2005, other technologies have surfaced, called **next-generation sequencing**. Since 2006, the Illumina (previously Solexa) technology is available and able to generate about 100 million reads per run on a single sequencing machine. These technologies produce shorter reads (anywhere from 25–500bp) but many hundreds of thousands or millions of reads in a relatively short time (on the order of a day) [8]. This results in high coverage, but the assembly process is much more computationally expensive. These technologies are vastly superior to shotgun sequencing due to the high volume of data and the relatively short time it takes to sequence a whole genome. The major disadvantage is that the accuracies are usually lower (although this is compensated for by the high coverage and read depth) [9].

1.1 Principle Methods (Past & Present)

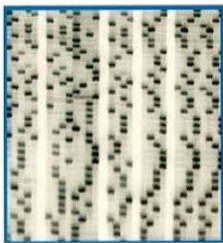





There are four principle methods for whole genome sequencing

- ❖ Sanger Sequencing Method (Whole genome shotgun sequencing)
- ❖ Hierarchical sequencing
- ❖ Pyrosequencing
- ❖ next-generation sequencing

1.1.1 Past Sequencing Methods

Sanger Sequencing Method (Whole genome shotgun sequencing) is the core principle methods used in past Sequencing technology during 1992- 2003 (figures of Instruments are given below)

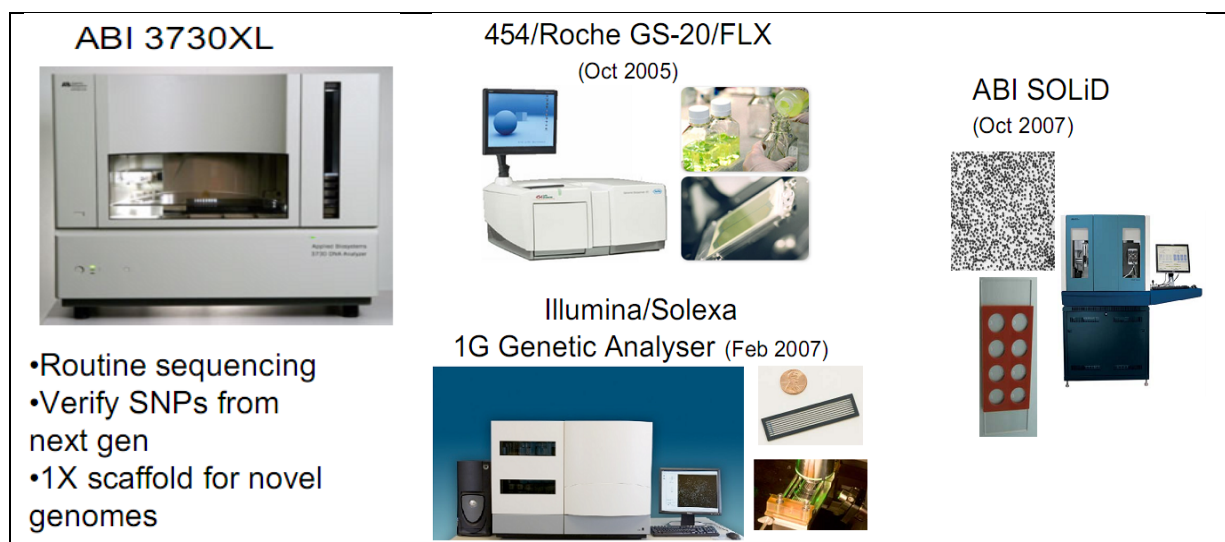
Pre-1992 "old fashioned way"	1992-1999 ABI 373/377	1999 ABI 3700	2003 ABI 3730XL
			
S35 ddNTPs Gels Manual loading Manual base calling	Fluorescent ddNTPs* Gels Manual loading Automated base calling*	Fluorescent ddNTPs Capillaries* Robotic loading* Automated base calling Breaks down frequently	Fluorescent ddNTPs Capillaries Robotic loading Automated base calling Reliable*

1.1.2 Present (Next Generation sequencing technology)

1. ABI 3730XL
2. 454/Roche GS-20/FLX
3. ABI SoLiD
4. Illumina/ Solexa Sequencing

The Illumina Sequencing is a core facility, which provides next-generation large-scale DNA (and RNA) sequencing services. Next-generation sequencing involves the application of glass microchip based methods and small-volume liquid handling (microfluidics) to sequence DNA more quickly and more cheaply than ever before, indeed about 100 times less costly than the technology used to sequence the first human genome just a few years ago. These methods rely on reacting millions of molecules simultaneously in a single vessel and analyzing those molecules in parallel on a single chip using a state-of-the-art optical detection instrument. A further increase in speed and a decrease in cost are attained by running multiple instruments

concurrently and Sanger Institute has 37 Solexa/Illumina DNA sequencing instruments available to tackle ambitious research projects in genomic medicine.



1.2 Comparison of NGS Technologies

Technology	Reads/run	Ave read length	bp per Run	Data output
3730XL (ABI)	96	900-1200 bp	~100,000	1-2MB
454 (Roche)	400,000	250-310 bp	70 million	20GB
Illumina 1G (Solexa)	40 million	36 bp	1 billion	1.5TB
SoLID (ABI)	88-132 million (44-66 per slide)	35 bp	1 billion	1.5-3.0TB

1.3 Next Generation Sequencing Steps

Next generation sequencing method – Illumina- Solexa 1G Genetic Analyzer was adopted for sequencing whole genome of *Phytophthora capsici*.

After sequencing, the sequencing platform generates sequencing images; and the data are analyzed in four steps: image analysis, base calling, sequence alignment, and variant analysis with counting (Figure 2) using CASAVA v1.7 (short for "Consensus Assessment of Sequence And VAriation") which is installed in supercomputer. Casava will give out short reads of nucleotide base pairs as output in Fastq format. This is then given to Maq software for assembly and variant detection using reference genome. Maq will give out `cns.snp` (to be converted to BED file) and indel files.

1. **Image analysis**— Uses the raw images to locate clusters, and outputs the cluster intensity, X, Y positions, and an estimate of the noise for each cluster. Firecrest method was used. The output from image analysis provides the input for base calling.
2. **Base calling**—Uses cluster intensities and noise estimates to output the sequence of bases read from each cluster, a confidence level for each base, and whether the read passes filtering. Base calling is performed by the instrument control software's Real Time Analysis (RTA) or the Off-Line Basecaller (OLB) using Bustard method.
3. **Sequence alignment**—Aligns samples to a reference sequence, GERALD alignment module was used. GERALD output is a flat text file containing each read and information about its alignment to the reference. Output is Fastq file which contain details about scaffolds and chromosomes.
4. **Variant analysis**— Maq software was used for mapping assemblies of scaffolds generated by NGS machines with reference genome to identify SNP's and indels. Maq software is particularly designed for Illumina- Solexa 1G Genetic Analyzer.

After variant analysis, the results can be viewed and analyzed further in third-party software or genome studio and has to be set up in genome view for annotation.

1.4 References

1. Staden R (June 1979). "A strategy of DNA sequencing employing computer programs". *Nucleic Acids Res.* **6** (7): 2601–10. doi:10.1093/nar/6.7.2601. PMC 327874. PMID 461197.
2. Roach, JC; Boysen, C; Wang, K; Hood, L (1995). "Pairwise end sequencing: a unified approach to genomic mapping and sequencing". *Genomics* **26** (2): 345–353. doi:10.1016/0888-7543(95)80219-C. PMID 7601461

3. Fleischmann, RD; et al. (1995). "Whole-genome random sequencing and assembly of *Haemophilus influenzae* Rd". *Science***269** (5223): 496–512. doi:10.1126/science.7542800. PMID 7542800.
4. Adams, MD; et al. (2000). "The genome sequence of *Drosophila melanogaster*". *Science***287** (5461): 2185–95. Bibcode2000Sci...287.2185.. doi:10.1126/science.287.5461.2185. PMID 10731132.
5. Dunham, I. *Genome Sequencing*. Encyclopedia of Life Sciences, 2005. doi: 10.1038/npg.els.0005378 .
6. Venter, J. C. "Shotgunning the Human Genome: A Personal View." Encyclopedia of Life Sciences, 2006.
7. Ronaghi, M., Uhlen, M., and Nyren, P. 1998. *Science* 281: 363. A sequencing method based on real-time pyrophosphate.
8. Karl, V; et al (2009). "Next Generation Sequencing: From Basic Research to Diagnostics". *Clinical Chemistry***55** (4): 41–47. doi:10.1373/clinchem.2008.112789. PMID 19246620.
9. Metzker, Michael L. (2010). "Sequencing technologies - the next generation". *Nat Rev Genet***11** (1): 31–46. doi:10.1038/nrg2626. PMID 19997069.

1.5 Further reading

- Jun Zhang , Rod Chiodini , Ahmed Badr , Genfa Zhang (2011), The impact of next-generation sequencing on genomics *Journal of Genetics and Genomics* 38 (2011) 95–109.

1.6 Whole Genome Sequencing and Informatics

Madavan Vasudevan, Bionivid Technology Private Limited

Genomics & Informatics

Bionivid
IT Complements Research

What is genomics?
Genomics is "the study of functions and interactions of all the genes in the genome, including their interactions with environmental factors." (Source: Collins, Francis, and Alan Guttmacher. "Genomic Medicine—A Primer," NEJM, Vol. 347:1512-1520.)

Genomics today !

- ✓ Agrigenomics
- ✓ Pharmacogenomics
- ✓ Epigenomics
- ✓ Metagenomics
- ✓ Nutrigenomics
- ✓ Personal Genomics

What is genome informatics?
It is computer based modeling and data processing of genome based data for studying various types of sequence and variation informatics, mostly in one way or another involving evolutionary analysis.
(Source: <http://www.sanger.ac.uk/research/projects/genomeinformatics/>)

Why to acquire genome informatics skills

Bionivid
IT Complements Research

Genomics and informatics are integral to the future of biotechnology and life sciences," — Kiran Mazumdar Shaw

Genomics = Informatics by Dr. Andras J. Pellionisz, International Hologenomics Society

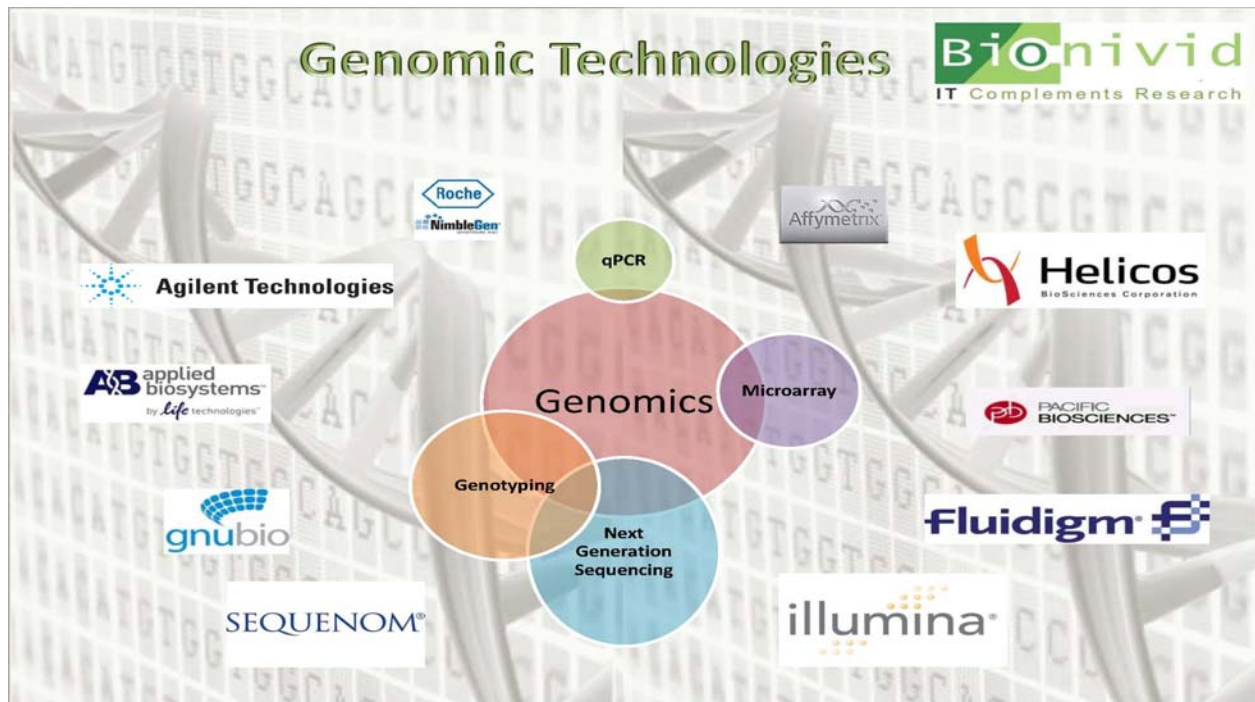
A quantitative understanding of biological function will only be achieved through development of structurally, biochemically, and biophysically detailed computational models based directly on experimental data. Once developed, these models can be simulated, analyzed, and understood through application of modern engineering and computational approaches, and the knowledge gained from these analyses can be applied to the design of additional experiments.

Raimond L. Winslow and Mark S. Boguski - The Johns Hopkins University School of Medicine and Whiting School of Engineering, Baltimore, Md; and the Human Biology Division (M.S.B.), Fred Hutchinson Cancer Research Center, Seattle,

Genomics = Informatics

Genomics Baby was tiny

The Genome Baby was left on the doorstep of IT

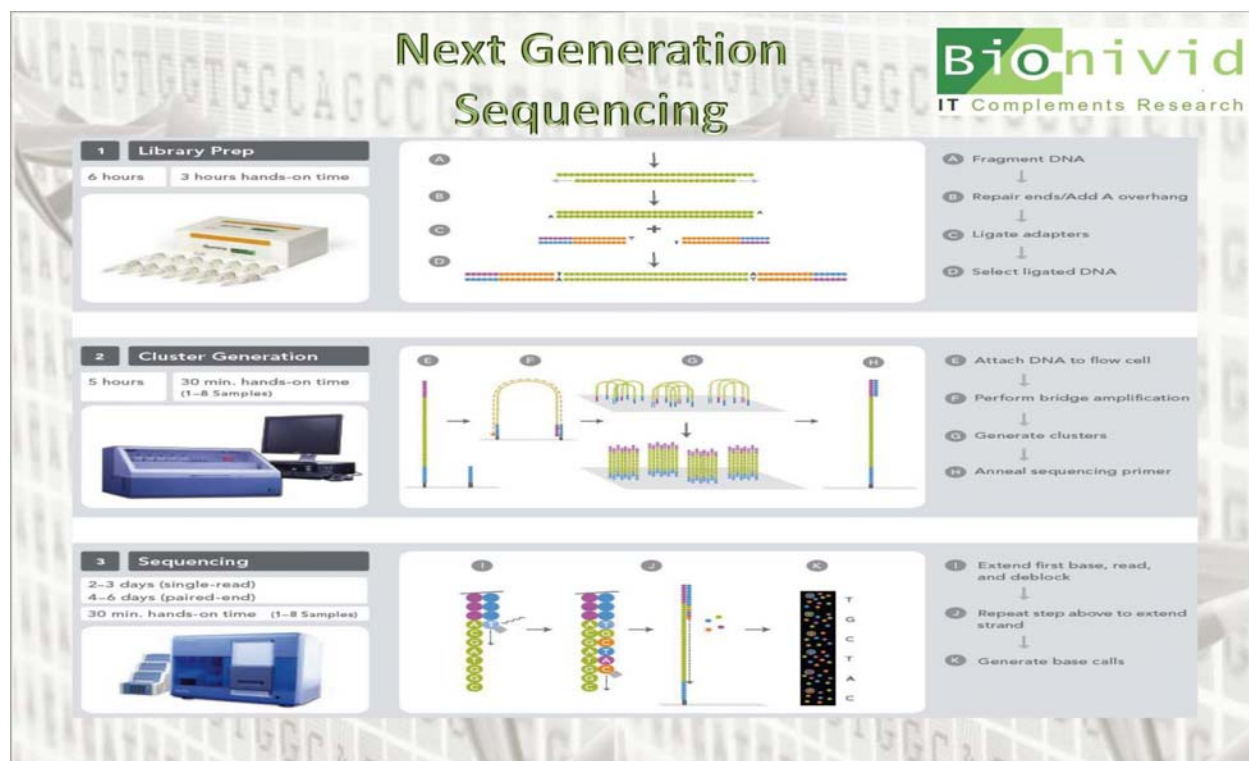


Introduction to Next Generation Sequencing (NGS) Technology

Bionivid
IT Complements Research

- The “original” sequencing methodology → Sanger chemistry → specifically labeled nucleotides to read through a DNA template during DNA synthesis.
- Sanger method has reached → 1000–1200 basepair (bp) → still cannot surpass 2 kilo basepair (Kbp)
- Shotgun sequencing → Human Genome Project → genomic DNA is enzymatically or mechanically broken down → cloned into sequencing vectors → sequenced individually
- Numerous fragments of DNA sequenced – BIRTH OF GENOME INFORMATICS AND NEXT GENERATION SEQUENCING
- The core philosophy of massive parallel sequencing used in next-generation sequencing (NGS) is adapted from shotgun sequencing
- NGS - breaking the entire genome into small pieces → ligating DNA to designated adapters → DNA synthesis (sequencing-by-synthesis) → massively parallel sequencing
- Coverage → number of short reads that overlap each other within a specific genomic region
- **Sufficient coverage is critical for accurate assembly of the genomic sequence.**
- To ensure the correct identification of genetic variants → short-read coverage → at least 30× is recommended in whole-genome scans

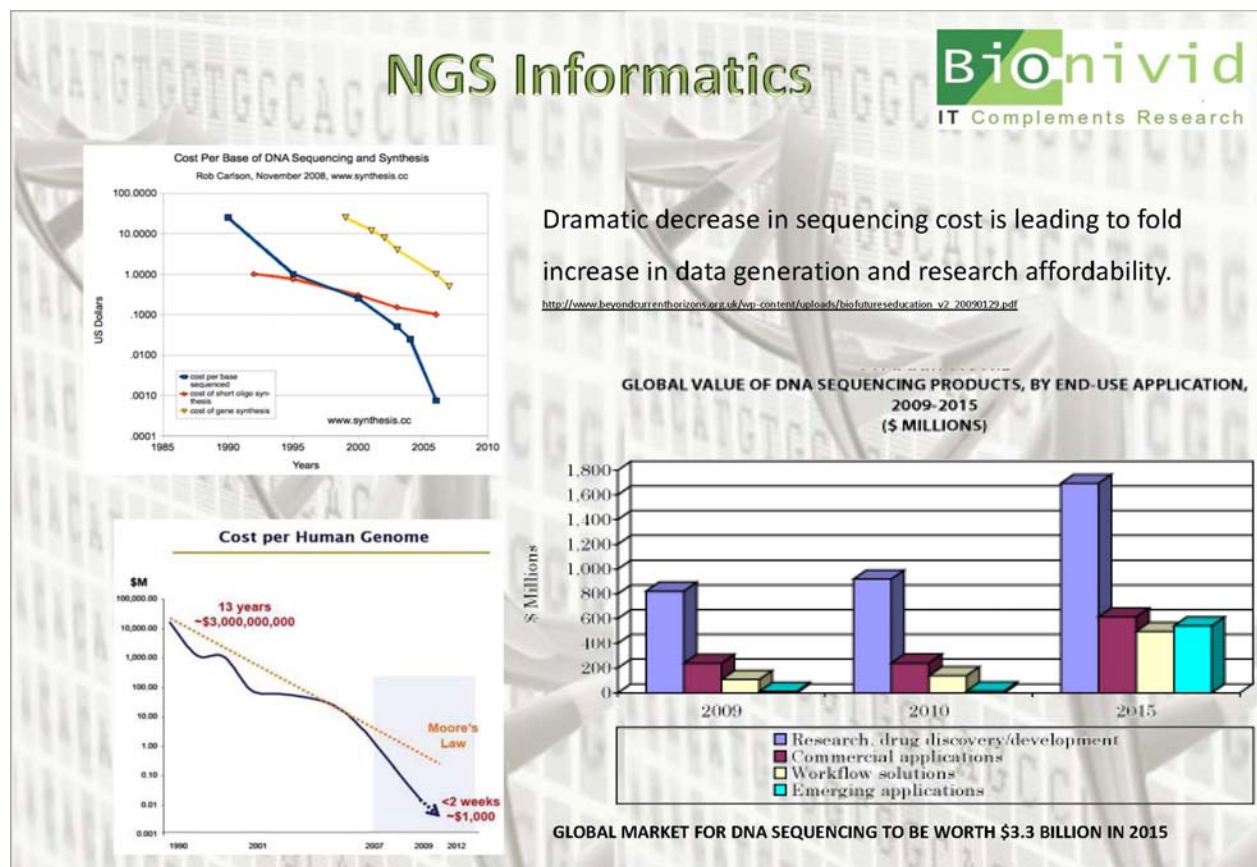
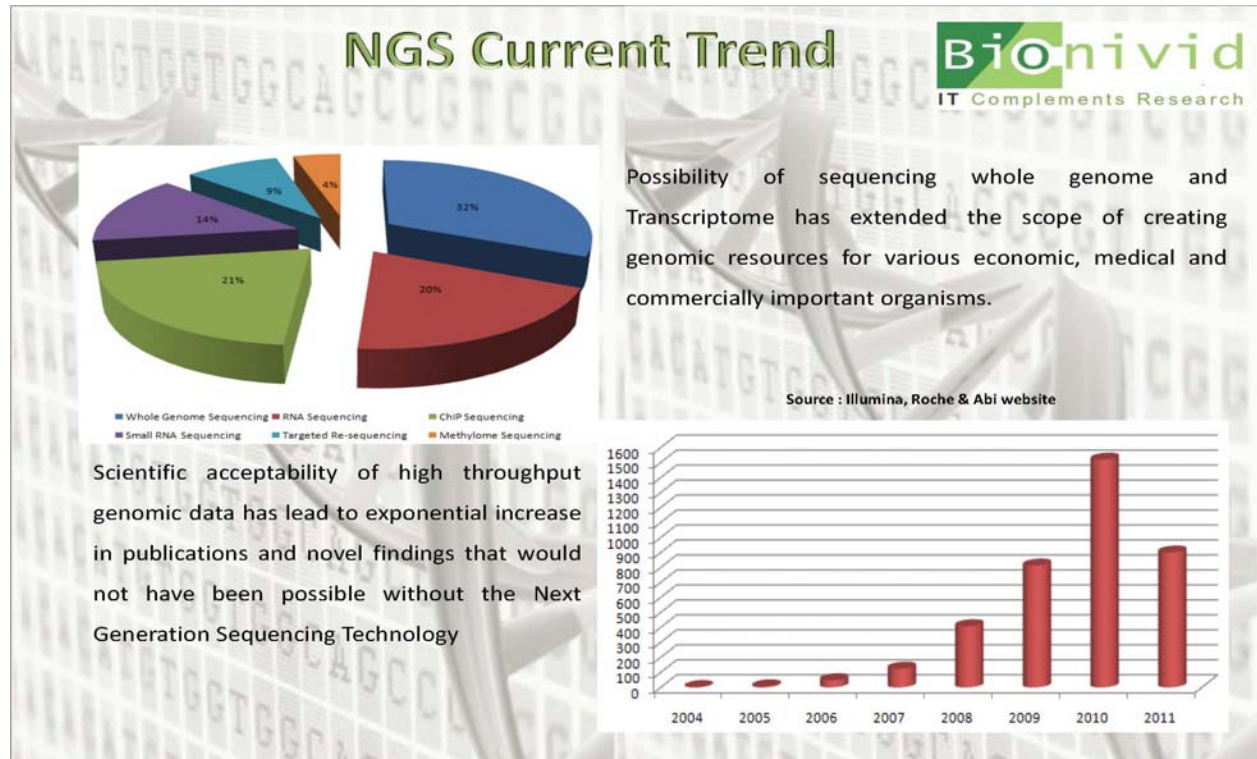
Short Clip

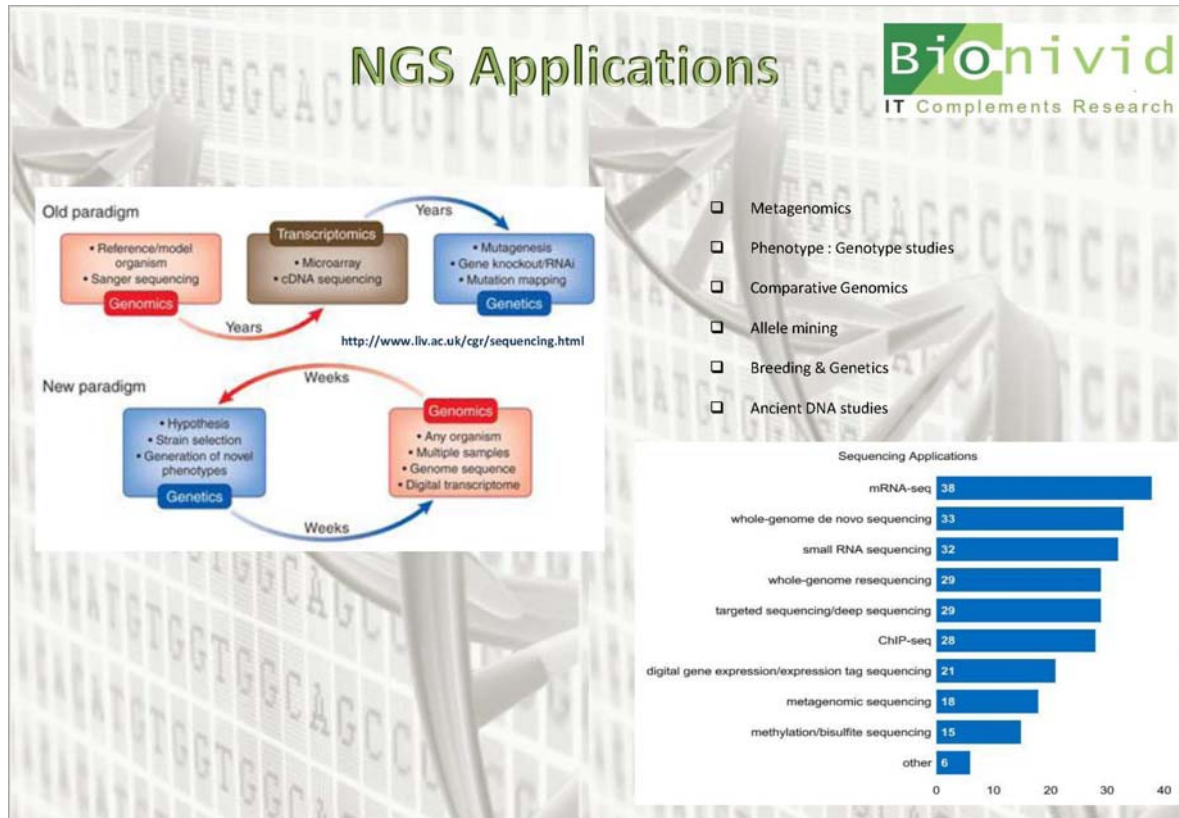


NGS Platforms

Bionivid
IT Complements Research

S.No	Platform	Amplification	Read Length	Throughput	Sequencing
1	Roche/GS-FLX Titanium	Emulsion PCR	400-600 bp	500 Mbp/run	Pyrosequencing
2	Illumina/HiSeq 2000, HiScan	Bridge PCR (Cluster PCR)	2 x 100 bp	200 Gbp/run	Reversible terminators
3	ABI/SOLID 5500xl	Emulsion PCR	50-100 bp	>100 Gbp/run	Sequencing-by-ligation (octamers)
4	Polonator/G.007	Emulsion PCR	26 bp	8-10 Gbp/run	Sequencing-by-ligation (monomers)
5	Helicos/Helioscope	No	35 (25-55) bp	21-37 Gbp/run	True single-molecule sequencing (tSMS)
	In development				
1	Pacific BioSciences/RS	No	1000 bp	N/A	Single-molecule real time (SMRT)
2	Visigen Biotechnologies	No	>100 Kbp	N/A	Base-specific FRET
3	U.S. Genomics	No	N/A	N/A	Single-molecule mapping
4	Genovox	No	N/A	N/A	Single-molecule sequencing by synthesis
5	Oxford Nanopore Technologies	No	35 bp	N/A	Nanopores/exonuclease-coupled
6	NABsys	No	N/A	N/A	Nanopores
7	Electronic BioSciences	No	N/A	N/A	Nanopores
8	BioNanomatrix/nanoAnalyzer	No	400 Kbp	N/A	Nanochannel arrays
9	GE Global Research	No	N/A	N/A	Closed Complex/nanoparticle
10	IBM	No	N/A	N/A	Nanopores
11	LingVitae	No	N/A	N/A	Nanopores
12	Complete Genomics	No	70 bp	N/A	DNA nanoball arrays
13	base4innovation	No	N/A	N/A	Nanostructure arrays
14	CrackerBio	No	N/A	N/A	Nanowells
15	Reveo	No	N/A	N/A	Nano-knife edge
16	Intelligent BioSystems	No	N/A	N/A	Electronics
17	LightSpeed Genomics	No	N/A	N/A	Direct-read Sequencing by EM
18	Halcyon Molecular	No	N/A	N/A	Direct-read Sequencing by EM
19	ZS Genetics	No	N/A	N/A	Direct-read Sequencing by TEM
20	Ion Torrent/PostLight	No	N/A	N/A	Semiconductor-based pH sequencing
21	Genizon BioSciences/CGA	No	N/A	N/A	Sequencing-by-hybridization





NGS Choice of Platform

Bionivid
IT Complements Research

Roche GS-FLX 454 Genome Sequencer

- At 600 bp, the 454 Sequencer has the longest short reads among all the NGS platforms; and generates ~400–600 Mb of sequence reads per run; critical for some applications such as *RNA isoform identification in RNA-seq and de novo assembly of microbes in metagenomics*. *Raw base accuracy reported by Roche is very good (over 99%)*

Illumina/Solexa Genome Analyzer

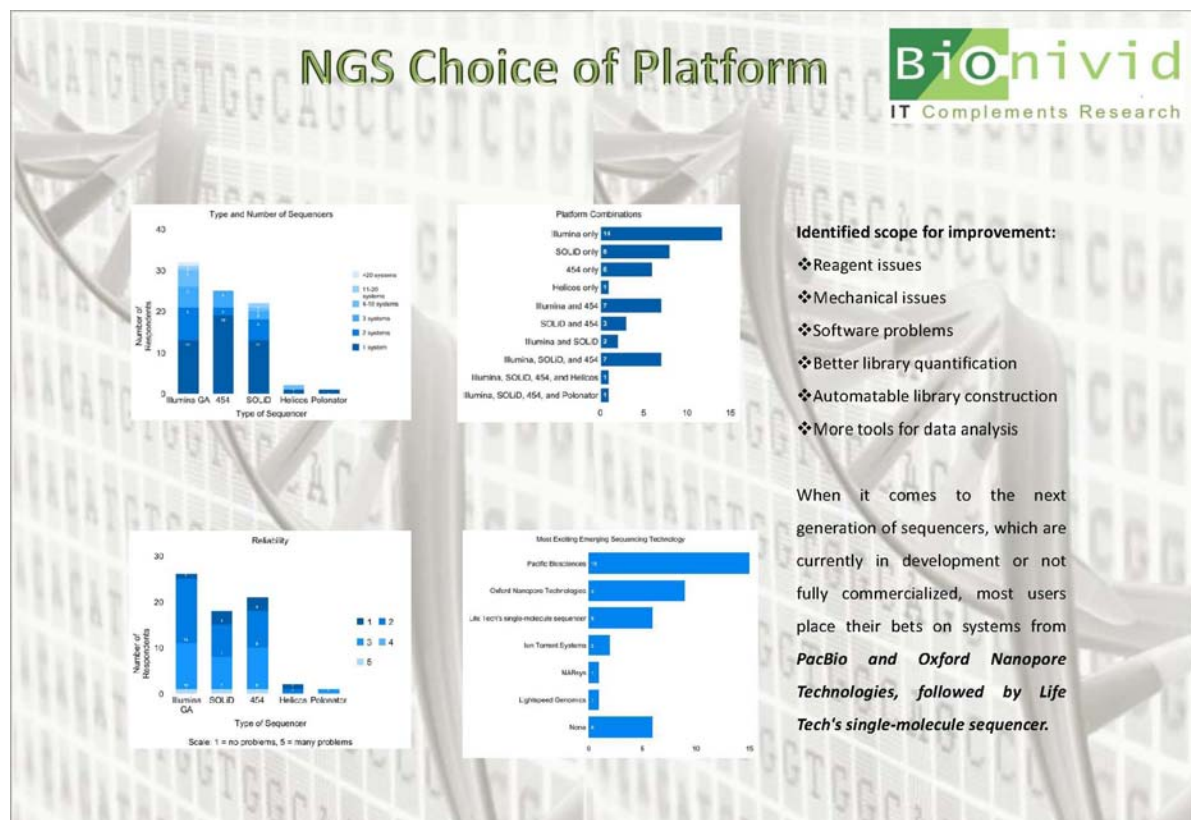
- Superior data quality and proper read lengths have made it the system of **choice for many genome sequencing projects**. To date, the *majority of published NGS papers* have described methods using the short sequence data produced with the Genome Analyzer. *The raw base accuracy is greater than 99.5%*

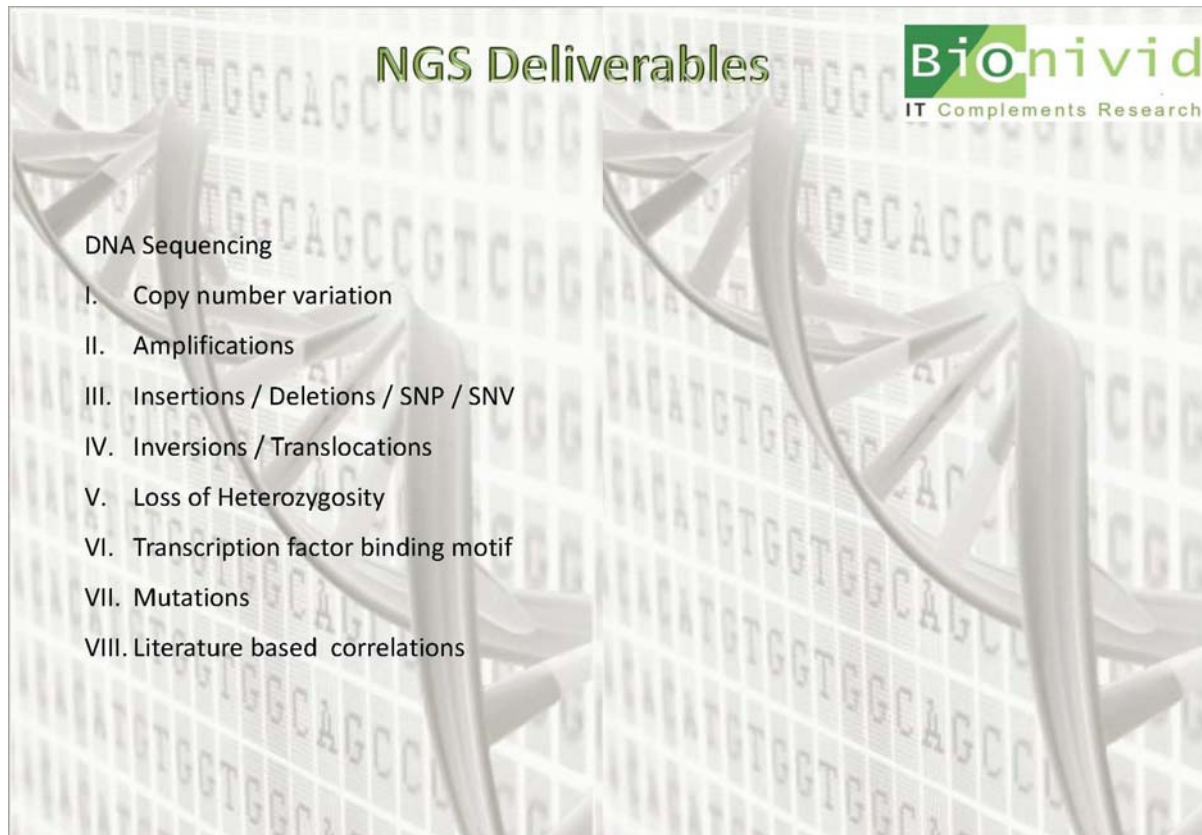
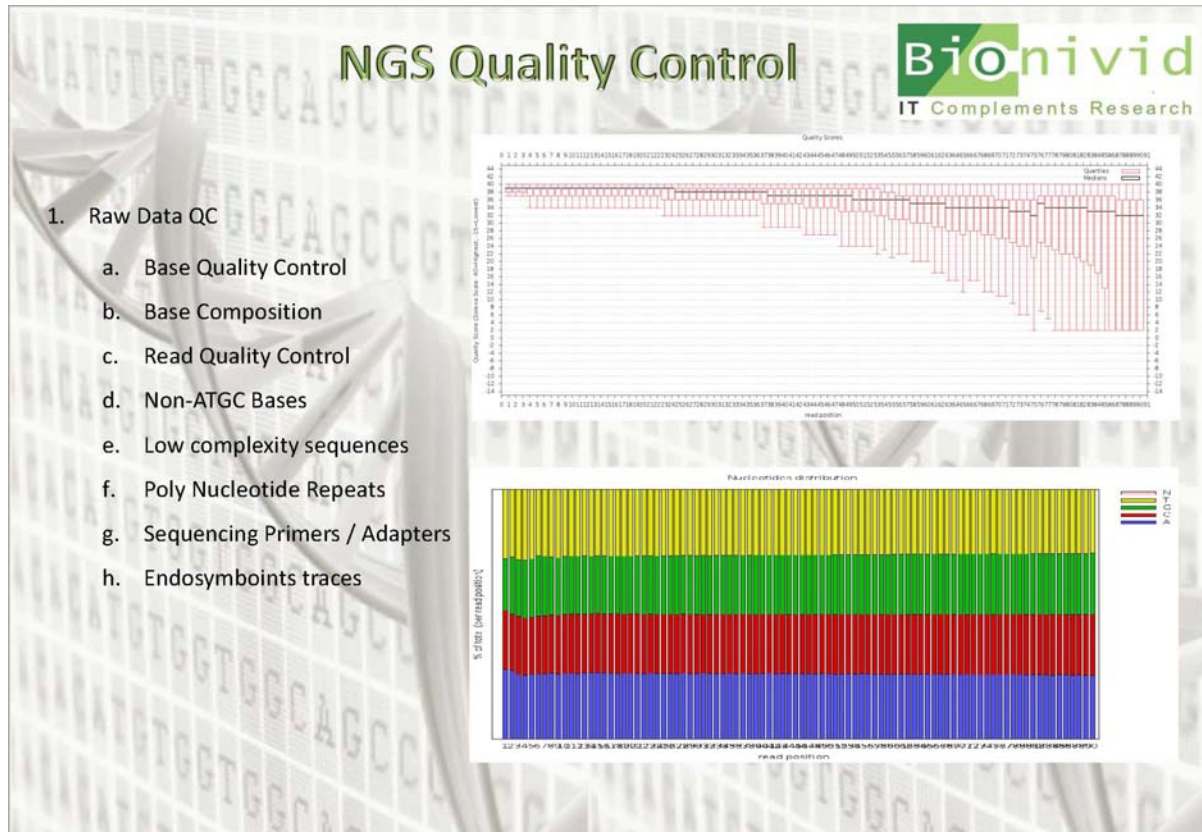
ABI SOLiD platform

- The latest model, 5500xl solid system (previously known as SOLiD4hq) can generate over 2.4 billion reads per run with a *raw base accuracy of 99.94%* due to its 2-base encoding mechanism. The SOLiD4 platform probably provides the best data quality as a result of its sequencing-by-ligation approach but the DNA library preparation procedures prior to sequencing can be tedious and time consuming. Preferred for Re-sequencing than DeNovo sequencing.

Emphasis on DeNovo Sequencing

Out of an amazing 514 projects, the majority of people preferred to use 454 for sequencing (286), about half as many used Illumina (144) and most of the rest went for a hybrid 454/Illumina approach. SOLiD (ABI) was used almost as much as Sanger, i.e. not a lot.





Page
14

2. Data Formats & Quality Check Analysis

2.1 SEQUENCE & READ DATA FORMATS

2.1.1 Next-Generation Sequencing Data

NGS data sets are very large, but as the cost per MB is now so low we expect to see them generated for many large-scale experiments where the mapping and analysis of the short read sets will be the key to interpretation of results.

For further analysis of NGS assembled data, whether a de novo assembly or using a set of NGS reads mapped to a reference sequence. Our initial aim is to rapidly produce a set of utility functions that can be used to read and interpret NGS assemblies, for example MAQ .map files, and to calculate statistics based on position, gene annotations, etc. These will then be used to build new applications and to demonstrate to end users, and to other developers, the ease of programming and the added value of the many interfaces already available.

NGS data volumes require new storage structures. As well as reading MAQ .map files and other formats for 454 sequencing and for new instruments) either by reading their formats or by using existing utilities to convert them. The instrument manufacturers show encouraging signs of using open data format and we will keep a close watch on emerging formats for a brief understanding

(i) NGS Data Sources

- NCBI
- EMBL
- DDBJ
- MINSEQE Minimum Information about a high-throughput Sequencing Experiment

(ii) Sequences

Sequences can be read and written in a variety of formats. These can be very confusing for users, but EMBOSS aims to make life easier by automatically recognizing the sequence format on input. That means that if you are converting from using another sequencing package to EMBOSS and you have your existing sequences in a format that is specific for that package, for example GCG format, you will have no problem reading them in. If you don't hold your sequence in a recognized standard format, you will not be able to analyze your sequence easily. Sequencers; read aligners; genome assemblers; scientific results from sequencing data; databases of reads; databases of genomes, variation, or other results. When all of these output data and require inputs in their own formats, we can be sure that there will be enormous amount of annoying, boring, and disappointing work while dealing with sequencing data.

(iii) What is sequence format?

Sequence formats are ASCII TEXT. They are the required arrangement of characters, symbols and keywords that specify what things such as the sequence, ID name, comments, etc. look like in the sequence entry and where in the entry the program should look to find them. There are generally no hidden, unprintable 'control' characters in any sequence format (there are none in those that EMBOSS supports). All standard sequence formats can be printed out or viewed simply by displaying their file.

(iv) Why so many formats?

There are at least a couple of dozen sequence formats in existence at the moment. Some are much more common than others. Formats were designed so as to be able to hold the sequence data and other information about the sequence. Nearly every sequence analysis package written since programs were first used to read and write sequences has invented its own format. Nearly every collection of sequences that dares call itself a database has stored its data in its own format. Most formats allow you to hold other description, annotation and comments like Fasta format. Other formats have specific fields for holding information such as references, keywords, associated entries in other databases and feature tables.

(v) Identification

Most sequence formats include at least one form of ID name, usually placed somewhere at the top of the sequence format. The simple format **FASTA** has the ID name as the first word on its title line. For example the ID name 'xyz':

```
>xyz some other comment
ttcctcttctcgactccatcttcgcggtagctgggaccgccgttcagtcgccaatatgc
agctctttgtccgcgccaggaagctacacaccttcgaggtgaccggccaggaacggtcg
cccagatcaaggctcatgtgcctcactggagggcatt
```

(vi) IDs and Accessions

An entry in a database must have some way of being uniquely identified in that database. Most sequence databases have two such identifiers for each sequence - an ID name and an Accession number.

Why are there two such identifiers? The ID name was originally intended to be a human-readable name that had some indication of the function of its sequence. In EMBL and GenBank the first two (or three) letters indicated the species and the rest indicated the function, for example 'hsfau' is the 'Homo Sapiens FAU pseudogene'. This naming scheme started to be a problem when the number of entries added each day was so vast that people could not make up

the ID names fast enough. Instead, the Accession numbers were used as the ID name. Therefore you will now find ID names like 'AF061303', the same as the Accession number for that sequence in EMBL. ID names are not guaranteed to remain the same between different versions of a database (although in practice they usually do).

Accession numbers are unique alphanumeric identifiers that are guaranteed to remain with that sequence through the rest of the life of the database. If two sequences are merged into one, then the new sequence will get a new Accession number and the Accession numbers of the merged sequences will be retained as 'secondary' Accession numbers. EMBL, GenBank and SwissProt share an Accession numbering scheme - an Accession number uniquely identifies a sequence within these three databases.

(vii) The Sequence

Nucleotide (DNA or RNA) sequences are usually stored in the IUBMB standard codes (International Union of Biochemistry and Molecular Biology).

Similarly, protein sequences are usually stored in the IUPAC standard one-letter codes (*International Union of Pure and Applied Chemistry*).

```
A = adenine
C = cytosine
G = guanine
T = thymine
U = uracil
R = G A (purine)
Y = T C (pyrimidine)
K = G T (keto)
M = A C (amino)
S = G C
W = A T
B = G T C
D = G A T
H = A C T
V = G C A
N = A G C T (any)
```

2.1.2 Sequence Database Formats

Some of the most widespread sequence formats apart from **fasta** are those used by the major sequence databases.

- EMBL
- GenBank
- SwissProt

- PIR

(i) Sequence Files

Files can hold sequences in standard recognized formats. Files can also hold sequences in non-standard unrecognizable ways. Do not save your sequences in a word-processor format file.

2.1.3 NGS Data Formats

To date, the following sequence formats are accepted as input for NGS tools.

(i) Read Formats (sample are given below)

- **FASTQ** format is a common format for short reads with quality scores. It is supported in EMBOSS 6.1.0 as a sequence format. Quality scores are also used if the format is more explicitly named in EMBOSS: fastqsanger or fastqillumina
- **SFF** Standard Flowgram Format, to hold the "trace" data for 454 reads
- **SRF** Sequence Read Format (also called Short Read Format), format specification has not been updated since February 2008. Applied Biosystems SRF Conversion Tool (solid2srf) converts SOLiD™ system reads into SRF format.
- **SCARF**, Solexa Compact ASCII Read Format. This format contains all information for one read in a single line. From left to right each line contains the read name, nucleotide sequence, quality scores for each position, and more information. Illumina's pipeline can produce SCARF files with quality scores in ASCII or numeric format.
- **SCF** first version was described in 1992, since then it has undergone several important changes such as a major reorganization of the ordering of the data items in the file and also in the way they are represented

(ii) Assembly Formats

- **MAQ .map** format (a compressed binary file format designed for short read alignment)
- **MAF**, MIRA Assembly Format
- **AMOS** A Modular Open-Source Assembler assembly format, used by velvet
- **SAM/BAM** (Sequence Alignment/Map) format is a generic format for storing large nucleotide sequence alignments

2.2 Sample Sequence and Read formats

2.2.1 Sequence formats

1. Plain sequence format

A sequence in plain format may contain only IUPAC characters and spaces (no numbers!).

Note: A file in plain sequence format may only contain one sequence, while most other formats accept several sequences in one file.

An example sequence in plain format is

```
ACAAGATGCCATTGTCCCCGGCCTCCTGCTGCTGCTGCTCTCCGGGGCCACGGCCACCGCT
GCCCTGCCCCTGGAGGGTGGCCCCACCGGCCGAGACAGCGAGCATATGCAGGAAGCGGCAG
GAATAAGGAAAAGCA
```

2. EMBL format

A sequence file in EMBL format can contain several sequences. One sequence entry starts with an identifier line ("ID"), followed by further annotation lines. The start of the sequence is marked by a line starting with "SQ" and the end of the sequence is marked by two slashes ("//").

An example sequence in EMBL format is:

```
ID  AB000263 standard; RNA; PRI; 368 BP.
XX
AC  AB000263;
XX
DE  Homo sapiens mRNA for preprocortistatin like peptide, complete cds.
XX
SQ  Sequence 368 BP;
acaagatgccattgtccccggcctcctgctgctgctgctctccggggccacggccaccg      60
ctgccctgcccctggaggggtggccccaccggccgagacagcgagcatatgcaggaagcgg      120
caggaataaggaagcagcctcctgacttctcgttggtggttgagtggaacctccc        180
aggccagtgcggggccctcataggagaggaagctcgggaggtggccaggcggcaggaag      240
gcgcacccccccagcaatccgcgcgccgggacagaatgccctgcaggaacttcttctgga      300
agaccttctcctcctgcaataaaacctcacccatgaatgctcacgcaagttaattaca      360
gacctgaa                                                              368
//
```

3. FASTA format

A sequence file in FASTA format can contain several sequences. Each sequence in FASTA format begins with a single-line description, followed by lines of sequence data. The description line must begin with a greater-than (">") symbol in the first column.

An example sequence in FASTA format is:

```
>AB000263 |acc=AB000263|descr=Homo sapiens mRNA for preprocortistatin like peptide,
complete cds.|len=368
ACAAGATGCCATTGTCCCCCGGCCTCCTGCTGCTGCTGCTCTCCGGGGCCACGGCCA
CCGCTGCCCTGCCCTGGAGGGTGGCCCCACCGGCCGAGACAGCGAGCATATGCAG
GAAGCGGCAGGAATAAGGAAAAGCAGC
```

4. GCG format

A sequence file in GCG format contains exactly one sequence, begins with annotation lines and the start of the sequence is marked by a line ending with two dot (".") characters. This line also contains the sequence identifier, the sequence length and a checksum. This format should only be used if the file was created with the GCG package.

An example sequence in GCG format is:

```
ID  AB000263 standard; RNA; PRI; 368 BP.
XX
AC  AB000263;
XX
DE  Homo sapiens mRNA for preprocortistatin like peptide, complete cds.
XX
SQ  Sequence 368 BP;
AB000263 Length: 368 Check: 4514 ..
1 acaagatgccattgtccccggcctcctgctgctgctctccggggccacggccaccg
61ctgccctgcccctggagggtggccccaccggccgagacagcgagcatatgcaggaagcgg
121caggaataaggaaaagcagcctcctgacttctcgttggtggttgagtgacacctcc
```

5. GCG-RSF (rich sequence format)

The new GCG-RSF can contain several sequences in one file. This format should only be used if the file was created with the GCG package.

6. GenBank format

A sequence file in GenBank format can contain several sequences. One sequence in GenBank format starts with a line containing the word LOCUS and a number of

annotation lines. The start of the sequence is marked by a line containing "ORIGIN" and the end of the sequence is marked by two slashes ("//").

An example sequence in GenBank format is:

```
LOCUS   AB000263           368 bp  mRNA  linear  PRI 05-FEB-1999
DEFINITION  Homo sapiens mRNA for preprocortistatin like peptide, complete
            cds.
ACCESSION  AB000263
ORIGIN
    1  acaagatgccattgtccccggcctcctgctgctgctctccggggccacggccaccg
    61  tgcctgcccctggagggtggccccaccggccgagacagcgagcatatgcaggaagcgg
   121  ggaataaggaaaagcagcctcctgactttctcgttggtggtttgagtggacctccc
   181  gccagtggcggggcccctcataggagaggaagctcgggaggtggccaggcggcaggaag
   241  gccccccccagcaatccgcgcggcgacagaatgccctgcaggaacttcttctgga
   301  acctctcctcctgcaaataaacctcacccatgaatgctcacgcaagttaattaca
   361  cctgaa
//
```

7. IG format

A sequence file in IG format can contain several sequences, each consisting of a number of comment lines that must begin with a semicolon (";"), a line with the sequence name (it may not contain spaces!) and the sequence itself terminated with the termination character '1' for linear or '2' for circular sequences.

An example sequence in IG format is:

```
; comment
; comment
AB000263
ACAAGATGCCATTGTCCCCCGGCCTCCTGCTGCTGCTGCTCTCCGGGGCCACGGCCA
CCGCTGCCCTGCC
CCTGGAGGGTGGCCCCACCGGCCGAGACAGCGAGCATATGCAGGAAGCGGCAGGA
ATAAGGAAAAGCAG1
```

2.2.2 Read Formats

Sequencers use their own output formats, and even multiple different formats per technology.

(i) Output formats from Roche 454 sequencer

.sff is 454's binary (=compressed) format, including the raw data. Not the very raw images, but flow values (flowgram), base calls, base-call qualities, read trimming and quality. There are many simple tools that “unzip” .sff into some readable textual format (see the black example).

.fna is a FASTA (or FASTA-like) format including the base-called reads.

```
>000007_1940_1402 length=172 uaccno=E4UQSRD01E0MP4
TAACAATCGAGGCGAAGTCCCGTGAGAAGCTGTTTACTTCTCATGATCACACAGGCGCTG
GCTCCTCAGGCAAACAGGTACGTCTACGATAGGTTCCATGAAAAGTCCAAGTTTGGCCGA
GCTCTGGCTCCTTTTGACGCACAGTGGAAGTTCCTTGTTACGGAAATTGCA
```

Figure 1: Read with .fna format.

.qual includes qualities corresponding to bases in .fna.

```
>000007_1940_1402 length=172 uaccno=E4UQSRD01E0MP4
28 35 28 27 34 27 26 25 25 28 31 24 26 27 32 25 27 27 32 28 6 28 27 27 27 27 33 26
27 26 27 27 34 30 10 27 25 34 27 28 22 28 27 26 26 27 27 26 27 25 22 23 28 27 18 20
23 27 27 29 21 25 25 34 26 27 24 25 32 24 22 33 28 7 25 20 30 22 28 27 24 25 28 28
28 27 28 26 27 25 23 33 25 35 28 34 27 27 25 28 38 34 21 8 25 27 34 27 31 23 22 36
32 17 29 21 32 24 24 27 28 19 27 28 26 34 28 23 25 35 28 38 34 21 8 26 26 27 25 27
21 28 28 27 27 34 27 34 27 25 30 21 34 26 33 25 26 35 28 20 28 25 34 27 37 33 15 33
25 23 28 25
```

Figure 2: Read with .qual format.

(ii) Output formats from Illumina

Illumina is over-creative with their own formats.

.seq.txt: base-called sequences

```
File: s_1_0001_seq.txt
1      1      137      689      AACATAATGTGTTCACTGAGAACACATTGCACTCAA
1      1      87       649      TATTGCAACTTGTTAATTTTTTCATGCCATTATCA
1      1      121      642      TACATGATTTGCATTTGGTAAATAGCTACTTTTAT
1      1      6       591      C...T.....T.....
```

Figure 3: Read with .seq.txt format.

.prb.txt: qualities for each nucleotide/each cycle (is some kind of semi-raw data?)

40	-40	-40	-40	40	-40	-40	-40	-40	40	-40	-40	-40	40	-40	-40	-40
-40	-40	-40	40	40	-40	-40	-40	-40	40	-40	-40	-40	-40	-40	-40	40
-40	-40	40	-40	-40	-40	-40	40	-40	-40	40	-40	-40	-40	-40	40	40
-40	-40	-40	40	-40	40	-40	-40	-40	40	-40	-40	-40	-40	-40	-40	-40
-40	-40	-40	40	-40	-40	40	40	-40	40	-40	-40	-40	-40	-40	40	-40
40	-40	-40	-40	40	-40	-40	-40	-40	-40	40	-40	-40	-40	40	-40	-40
-40	40	-40	-40	40	-40	-40	-40	-40	-40	-40	-40	-40	40	-40	-40	40
-40	-40	40	-40	-40	40	-40	-40	-40	40	-40	-40	-40	-40	40	-40	-40
-40	-40	-40	40	-40	40	-40	-40	-40	40	-40	-40	-40	-40	40	-40	-40
37	-37	-40	-40													

Figure 4: Read with .prb.txt format.

Illumina FASTQ is not real FASTQ, but looks similarly (to cause confusion). The quality score is an own Illumina score, computed by subtracting 64 from the ASCII code of the character. For example the 'h' in the example is $\text{ord}('h') - 64 = 104 - 64 = 40$.

```
@ILMN-GA001 3 208HWAAXX 1 1 110 812
ATACAAGCAAGTATAAGTTCGTATGCCGTCTT
+ILMN-GA001 3 208HWAAXX 1 1 110 812
hhhYhh]NYhhhhhhYIhhaZT[hYHNSPKXR
@ILMN-GA001 3 208HWAAXX 1 1 111 879
GGAGGCTGGAGTTGGGGACGTATGCGGCATAG
+ILMN-GA001 3 208HWAAXX 1 1 111 879
hSWHRNJ\hFhLdhVOhAIB@NFKD@PAB?N?
```

Figure 5: Read with .fastq format.

See <http://www.asciitable.com/> for ASCII codes (Dec is the 'ord' in decimal numbers).

The Illumina score can be transformed to Phred score. Given a character *ch*, the following expression should give the Phred quality *q* (please double-check:) $q = 10 * \log(1 + 10^{(\text{ord}(\text{ch}) - 64) / 10.0}) / \log(10)$

Qseq: Not looking like FASTQ, but the scores are proper Phred scores. Their encoding is however still ASCII - 64.

```
HWUSI-EAS521 2 1 26 0 76 0 1
.GGCAGCGGGCAGGGCGAGCCAATGCGTGTGGGGGGGGGGGGCTCGCAGTGGGGGGGGAACGGCGAGTGCGGGGGG
BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB 0
HWUSI-EAS521 2 1 26 0 562 0 1
.GGGAAATAGCTTTCACGCCTTTAGATAATTTCAATAAAATCATAGCGCCAATGGGGAGCAAACCTACCATACACC
BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB 0
HWUSI-EAS521 2 1 26 0 1113 0 1
.ATCTTTAACAGACCAAGACTGGGCCACAAGCCTCCAGACTGTAACTGCTTCAAAGAGGCTTAGGCAGGCAGG
BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB 0
HWUSI-EAS521 2 1 26 0 244 0 1
.GGCTGGGTATGAGTCAGGGGGCTCCAGAGAGACAGAACCAGTCGGACATCGACAGATAGATCGGGGGGGAGTTAT
BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB 0
```

Figure 6: Read with .Qseq format.

Illumina “single line format” and SCARF: base-called sequences and quality scores in a bit more handy format.

```
>1-1-137-689    AACATAATGTTCACTGAGAACACATTGCACTCAA    U0
>1-1-87-649     TATTGCAACTGTTTAATTTTTCATGCCATTATCA    U1
>1-1-121-642    TACATGATTGTCATTGGTAAATAGCTACTTTTAT    U0
```

Figure 7: Read with single line format

```
HWI-EAS102_3 : 6 : 1 : 897 : 791 : AATGTCAATCTGAGTT ... TTT : 40 40 40 40 40 ..
HWI-EAS102_3 : 6 : 1 : 930 : 291 : AATGTACTTTTCTAA ... CTA : 40 29 14 17 16 ..
HWI-EAS102_3 : 6 : 1 : 944 : 665 : AATCGATCCCCTTCCC ... TTC : 40 34 33 40 40 ..
```

Figure 8: Read with SCARF format

(iii) Output formats from SOLiD

SOLiD outputs **CSFASTA**. It's a semi-raw format similar to FASTQ. Read starts with the first base, but colour calls follow.

```
@ERR000451.1 VAB_S0103_20080915_542_14_17_70_F3
T33023230203102103223330020300233001
+
T%245719<.6353&:%0#%&%2(--27*%&%,
```

Figure 9: Read with CsFASTA format

2.2.3 The two main public read archives and their two main formats.

It's however crucial for each data set to decide what data should be deleted, and after what time! When and whether to delete the raw data, flows, colours, when and whether the base-called reads & qualities, alignments. The data is often enormously huge and it will cost you and/or the national/global institutes a lot of money and energy to store and back them up. Hence we use SRA format (XML), SRF format.

(i) Assembly Format Sample

Read aligners output a number of more or less similar own formats. However many of them output data in the common **SAM/BAM format**, what is good news.

SAM is a plain-textual format of the alignments (in a flavour that is similar but different to **GFF** or **BED**). It is luckily extensible.

BAM is a dedicated binary format including the compressed SAM. It enables fast access to data without having to “unzip” the whole file.

For the typically large data, BAM is currently the most recommended and most “standard” format.

Figure 10: Read with SAM format

BioXSD is a new set of structured, “object-oriented” formats for exchange of sequence data, any kind of sequence/genome annotation, and related. It is under development and optimised for use with (but not limited to) Web services, and aims to becoming a standard format in future. Very generic and extensible thanks to the use of XML Schema (XSD).

Figure 11: Read with SAM format

Nucleotide sequence databases (GenBank, EMBL, DDBJ, gathered under the INSDC consortium) store a number of types of metadata about the genomic sequences.

However, additional metadata is crucial to enable efficient future research using the submitted genome & metagenome-sequencing data. GSC consortium has therefore defined the MIGS and MIMS standards for additional metadata that should be submitted to a public repository.

“The minimum information about a genome sequence (MIGS) specification” .Extended to the Minimum Information about a Metagenomic Sequence/Sample: MIMS. MIENS (environmental sequencing) follow. GCDML is an XML format for MIGS/MIMS/MIENS metadata, (Using XML Schema and SAWSDL annotations.)

(iv) Genomatix annotation syntax

Some Genomatix tools, e.g. Gene2Promoter or GPD allow the extraction of sequences. Genomatix uses the following syntax to annotate sequence information: each information item is denoted by a keyword, followed by a "=" and the value. These information items are separated by a pipe symbol "|".

The keywords are the following:

loc	The Genomatix Locus Id , consisting of the string "GXL_" followed by a number.
sym	The gene symbol . This can be a (comma-separated) list.
geneid	The NCBI Gene Id . This can be a (comma-separated) list.
acc	A unique identifier for the sequence. E.g. for Genomatix promoter regions, the Genomatix Promoter Id is listed in this field.
taxid	The organism's Taxon Id
spec	The organism name
chr	The chromosome within the organism.
ctg	The NCBI contig within the chromosome.
str	Strand , (+) for sense, (-) for antisense strand.
start	Start position of the sequence (relative to the contig).
end	End position of the sequence (relative to the contig).
len	Length of the sequence in basepairs.
tss	A (comma-separated list of) UTR-start/TSS position(s) . If there are several TSS/UTR-starts, this means that several transcripts share the same promoter (e.g. when they are splice variants). The positions are relative to the promoter region.
probe	A (comma-separated list of) Affymetrix Probe Id(s) .
unigene	A (comma-separated list of) UniGene Cluster Id(s) .
homgroup	An identifier (a number) for the homology group (available for promoter sequences only). Orthologously related sequences have the same value in this field.
promset	If the sequence is a promoter region, the promoter set is denoted here.
descr	The gene description . If several genes (i.e. NCBI gene ids) are associated with the

	sequence, the descriptions for all of the genes are note, separated by ";"
comm	A comment field, used for additional annotation. For promoter sequences, this field contains information about the transcripts associated with the promoter. For each transcript the Genomatix Transcript Id, accession number, TSS position and quality is listed, separated by "/". For GenomatixCompGen promoters no transcripts are assigned, in this case the string "CompGen promoter" is denoted.

2.3 Quality Check

Next generation sequencing (NGS) technologies provide a high-throughput means to generate large amount of sequence data. However, quality check (QC) of sequence data generated from these technologies is extremely important for meaningful downstream analysis. Further, highly efficient and fast processing tools are required to handle the large volume of datasets. Modern high throughput sequencers can generate tens of millions of sequences in a single run. Before analyzing this sequence to draw biological conclusions you should always perform some simple quality control checks to ensure that the raw data looks good and there are no problems or biases in your data which may affect how you can usefully use it.

Most sequencers will generate a QC report as part of their analysis pipeline, but this is usually only focused on identifying problems which were generated by the sequencer itself. QC aims to get a QC report which can spot problems which originate either in the sequencer or in the starting library material.

After obtaining reads (Fast Q, Fast A) from sequencer machine, first step is to check for QC data

- ❖ Next generation high throughput sequencers generate huge amount of sequencing data (100Mb to few GB's).
- ❖ Quality check and primary analysis of raw sequence data is vital prior to the in-depth analysis.
- ❖ QC and primary analysis of the huge sequencing data using ordinary methods / machines is close to impossible.

2.4 Quality Check analysis using Script

QC analysis of Raw Read Data and removing the redundancy (filtering)

QC analysis is done to identify low complexity regions, depending upon the QC data we can decide whether our data is correct or not. Based on the QC parameters, reads are filtered out. Only those sequence reads which are good enough will be taken out. Sequence depth is completely based on how much filtered quality reads obtained. here we are doing QC for read 1 and read 2 independently in first two steps, running raw data Qc will give out 2 type of result-

filtered data and failed data, filtered data will be the reads which pass parameters of QC (output-filtered_reads.txt). Finally removing the redundancy in 3rd step, removing redundant reads will obtain filtered mapped reads from overlapping reads, output is (filtered_matched.txt).

1.QC for read1	sh ./Sequence_QC.sh Read_1.fastq
2.QC for read2	sh ./Sequence_QC.sh Read_2.fastq
3.QC for read1 and read2 to filter out redundancy or common reads	sh./filteredCommonPereads.shRead_1_filtered.fastq Read_2_filtered.fastq

Output

This will give out six types of output for each sequence reads that will help to judge the quality.

- Readname_filtered.fastq
- Readname_ndGraph.png
- Readname_qbGraph.png
- Readname_QC_report.txt
- Readname_Qstats.txt
- Readname_sorted_tab.txt

2.5 Quality Check analysis using Tool: FastQC

FastQC aims to provide a QC report which can spot problems which originate either in the sequencer or in the starting library material. FastQC can be run in one of two modes. It can either run as a standalone interactive application for the immediate analysis of small numbers of FastQ files, or it can be run in a non-interactive mode where it would be suitable for integrating into a larger analysis pipeline for the systematic processing of large numbers of files.

Steps:

2.5.1 Basic Operations

(i) Opening a Sequence file

To open one or more Sequence files interactively simply run the program and select File > Open. You can then select the files you want to analyze.

Newly opened files will immediately appear in the set of tabs at the top of the screen. Because of the size of these files it can take a couple of minutes to open them. FastQC operates a queuing system where only one file is opened at a time, and new files will wait until existing files have been processed.

FastQC supports files in the following formats

- FastQ (all quality encoding variants)
- CasavaFastQ files*
- ColospaceFastQ
- GZip compressed FastQ
- SAM
- BAM
- SAM/BAM Mapped only (normally used for color space data)

* Casavafastq format is the same as regular fastq except that the data is usually split across multiple files for a single sample. In this mode the program will merge the files in a sample group and present a single report for each sample. Also Casavafastq files contain poor quality sequences which have been flagged to be removed. In Casava mode the program will exclude these flagged sequences from the report.

By default FastQC will try to guess the file format from the name of the input file. Anything ending in .sam or .bam will be opened as a SAM/BAM file (using all sequences, mapped and unmapped), and everything else will be treated as FastQ format. If you want to override this detection and specify the file format manually then you can use the drop down file filter in the file chooser to select the type of file you're going to load. You need to use the drop down selector to make the program use the Mapped BAM or Casava file modes as these won't be selected automatically.

(ii) Evaluating Results

The analysis in FastQC is performed by a series of analysis modules. The left hand side of the main interactive display or the top of the HTML report show a summary of the modules which were run, and a quick evaluation of whether the results of the module seem entirely normal (green tick), slightly abnormal (orange triangle) or very unusual (red cross).

It is important to stress that although the analysis results appear to give a pass/fail result, these evaluations must be taken in the context of what you expect from your library. A 'normal' sample as far as FastQC is concerned is random and diverse. Some experiments may be expected to produce libraries which are biased in particular ways. You should treat the summary evaluations

therefore as pointers to where you should concentrate your attention and understand why your library may not look random and diverse.

(iii) Saving a Report

In addition to providing an interactive report FastQC also has the option to create an HTML version of this report for a more permanent record. This HTML report can also be generated directly by running FastQC in non-interactive mode.

To create a report simply select File > Save Report from the main menu. By default a report will be created using the name of the fastq file with _fastqc.zip appended to the end. The report will be created for whichever file tab was active when the menu option was selected.

The report file which is created is actually a zip file. When uncompressed this will create a folder with the same name as the report file. Inside this will be a series of files, but the one to look at is the fastqc_report.html file which links to the other files in the report.

2.5.2 Analysis modules

(i) Basic Statistics

The Basic Statistics module generates some simple composition statistics for the file analyzed.

- **Filename:** The original filename of the file which was analyzed
- **File type:** Says whether the file appeared to contain actual base calls or color space data which had to be converted to base calls
- **Encoding:** Says which ASCII encoding of quality values was found in this file.
- **Total Sequences:** A count of the total number of sequences processed. There are two values reported, actual and estimated. At the moment these will always be the same. In the future it may be possible to analyze just a subset of sequences and estimate the total number, to speed up the analysis, but since we have found that problematic sequences are not evenly distributed through a file we have disabled this for now.
- **Filtered Sequences:** If running in Casava mode sequences flagged to be filtered will be removed from all analyses. The number of such sequences removed will be reported here. The total sequences count above will not include these filtered sequences and will be the number of sequences actually used for the rest of the analysis.
- **Sequence Length:** Provides the length of the shortest and longest sequence in the set. If all sequences are the same length only one value is reported.
- **%GC:** The overall %GC of all bases in all sequence

(ii) Per Base Sequence Quality

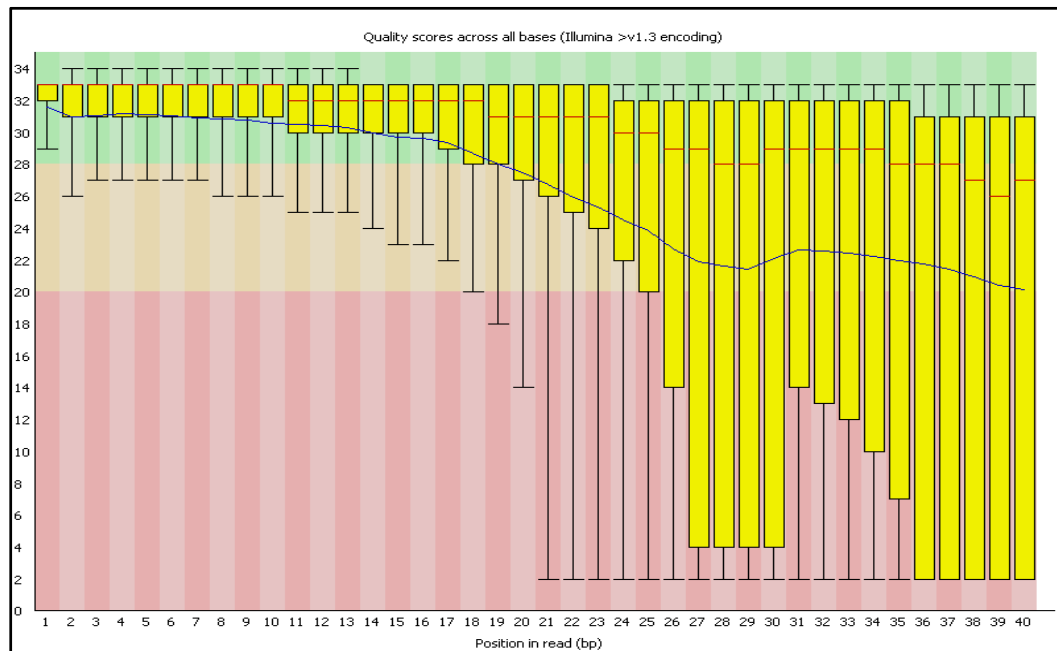


Figure 1: This view shows an overview of the range of quality values across all bases at each position in the FastQ file.

For each position a Box Whisker type plot is drawn. The elements of the plot are as follows:

- The central red line is the median value
- The yellow box represents the inter-quartile range (25-75%)
- The upper and lower whiskers represent the 10% and 90% points
- The blue line represents the mean quality

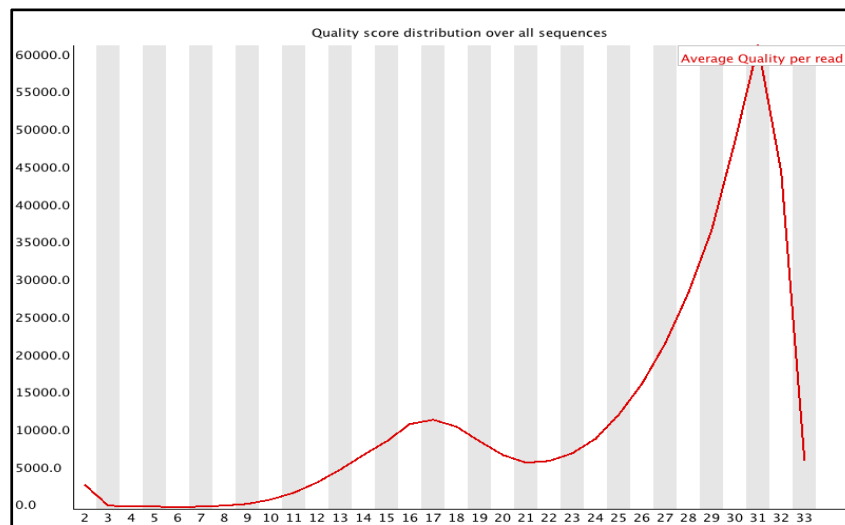
The y-axis on the graph shows the quality scores. The higher the score the better the base call. The background of the graph divides the y axis into very good quality calls (green), calls of reasonable quality (orange), and calls of poor quality (red). The quality of calls on most platforms will degrade as the run progresses, so it is common to see base calls falling into the orange area towards the end of a read.

It should be mentioned that there are number of different ways to encode a quality score in a Fast Q file. FastQC attempts to automatically determine which encoding method was used, but in some very limited datasets it is possible that it will guess this incorrectly (ironically only when your data is universally very good!). The title of the graph will describe the encoding FastQC thinks your file used.

A warning will be issued if the lower quartile for any base is less than 10, or if the median for any base is less than 25. This module will raise a failure if the lower quartile for any base is less than 5 or if the median for any base is less than 20.

(iii) Per Sequence Quality Scores

The per sequence quality score report allows you to see if a subset of your sequences have universally low quality values. It is often the case that a subset of sequences will have universally poor quality, often because they are poorly imaged (on the edge of the field of view etc), however these should represent only a small percentage of the total sequences.

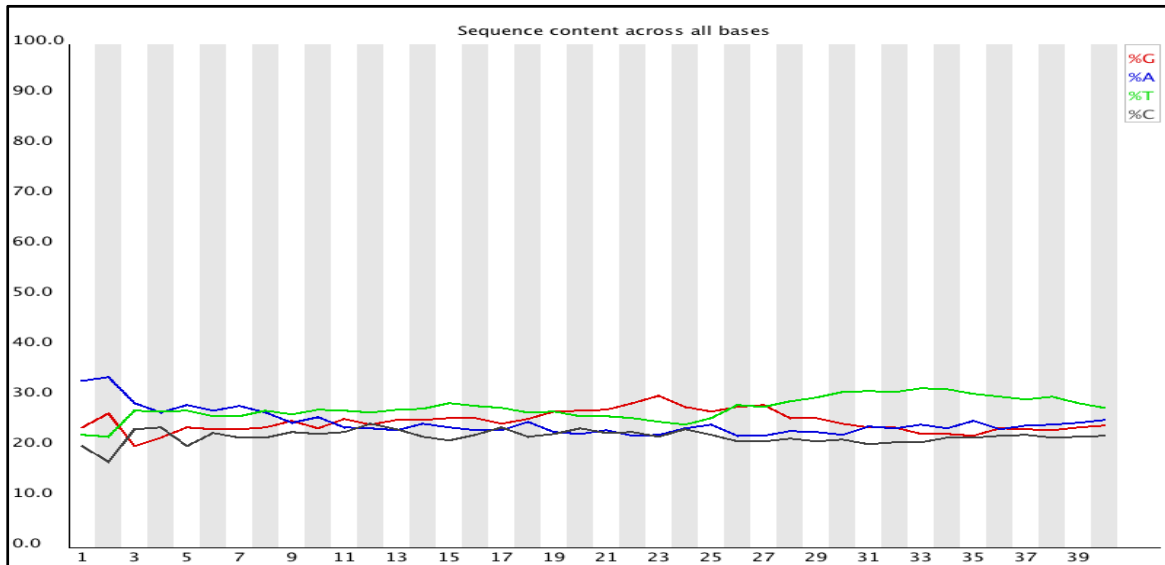


If a significant proportion of the sequences in a run have overall low quality then this could indicate some kind of systematic problem - possibly with just part of the run (for example one end of a flowcell).

A warning is raised if the most frequently observed mean quality is below 27 - this equates to a 0.2% error rate. An error is raised if the most frequently observed mean quality is below 20 - this equates to a 1% error rate.

(iv) Per Base Sequence Content

Per Base Sequence Content plots out the proportion of each base position in a file for which each of the four normal DNA bases has been called.



In a random library you would expect that there would be little to no difference between the different bases of a sequence run, so the lines in this plot should run parallel with each other. The relative amount of each base should reflect the overall amount of these bases in your genome, but in any case they should not be hugely imbalanced from each other.

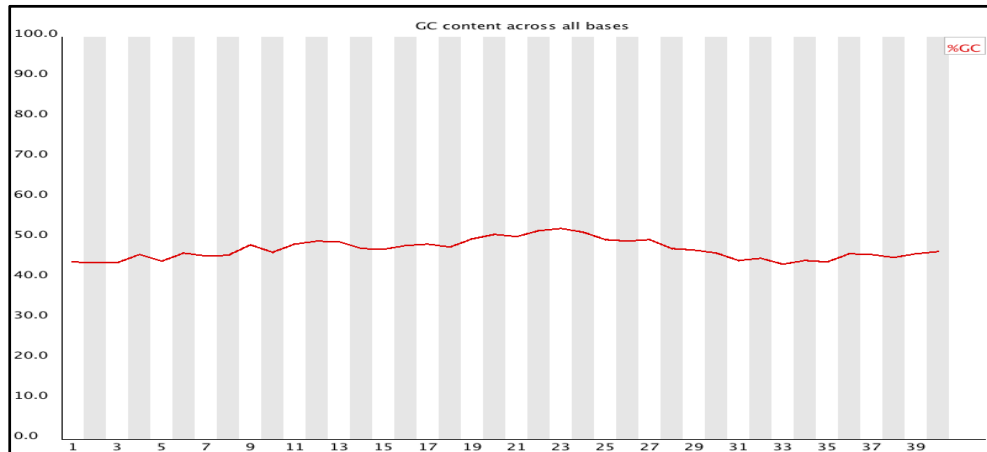
If you see strong biases which change in different bases then this usually indicates an overrepresented sequence which is contaminating your library. A bias which is consistent across all bases either indicates that the original library was sequence biased, or that there was a systematic problem during the sequencing of the library.

This module issues a warning if the difference between A and T, or G and C is greater than 10% in any position. This module will fail if the difference between A and T, or G and C is greater than 20% in any position.

(v) Per Base GC Content

Per Base GC Content plots out the GC content of each base position in a file.

In a random library you would expect that there would be little to no difference between the different bases of a sequence run, so the line in this plot should run horizontally across the graph. The overall GC content should reflect the GC content of the underlying genome.

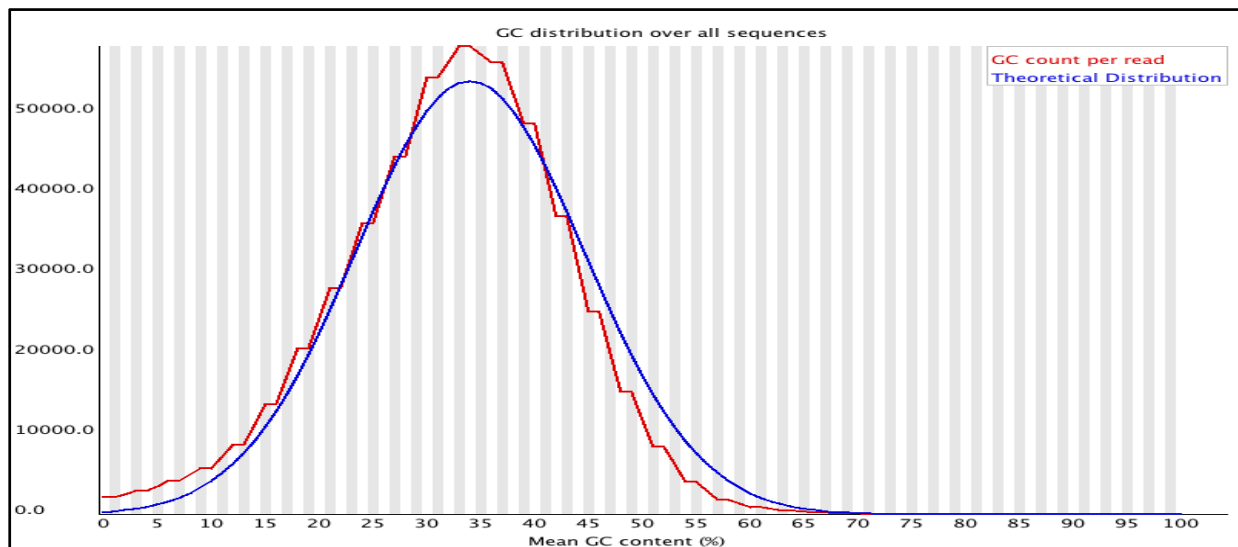


If you see a GC bias which changes in different bases then this could indicate an over presented sequence which is contaminating your library. A bias which is consistent across all bases either indicates that the original library was sequence biased, or that there was a systematic problem during the sequencing of the library.

This module issues a warning if the GC content of any base strays more than 5% from the mean GC content. This module will fail if the GC content of any base strays more than 10% from the mean GC content.

(vi) Per Sequence GC Content

This module measures the GC content across the whole length of each sequence in a file and compares it to a modeled normal distribution of GC content.



In a normal random library you would expect to see a roughly normal distribution of GC content where the central peak corresponds to the overall GC content of the underlying genome. Since

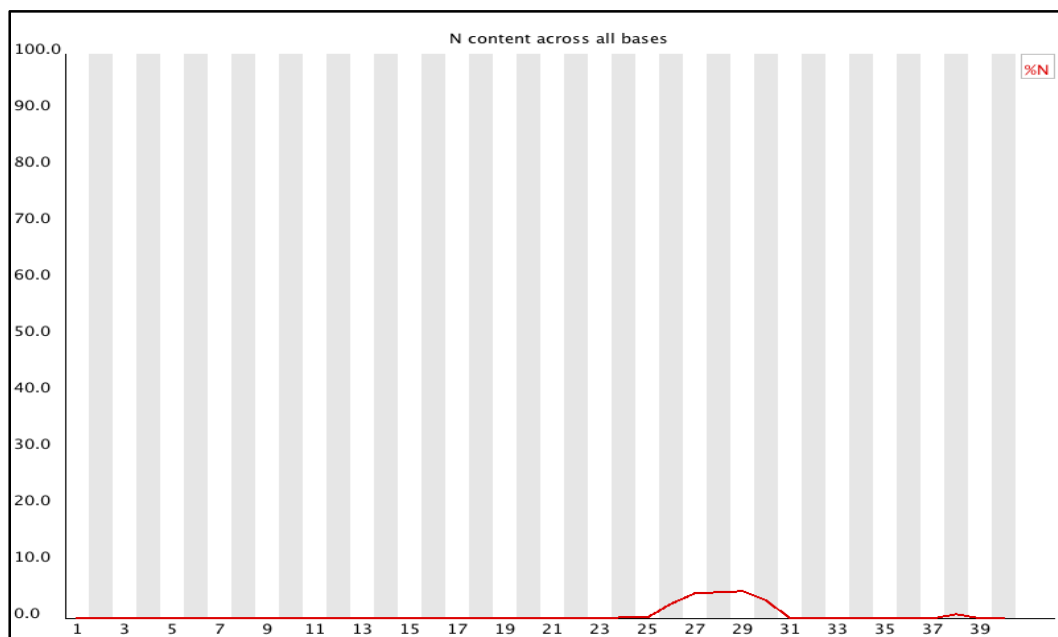
we don't know the the GC content of the genome the modal GC content is calculated from the observed data and used to build a reference distribution.

An unusually shaped distribution could indicate a contaminated library or some other kinds of biased subset. A normal distribution which is shifted indicates some systematic bias which is independent of base position. If there is a systematic bias which creates a shifted normal distribution then this won't be flagged as an error by the module since it doesn't know what your genome's GC content should be.

A warning is raised if the sum of the deviations from the normal distribution represents more than 15% of the reads. This module will indicate a failure if the sum of the deviations from the normal distribution represents more than 30% of the reads.

(vii) Per Base N Content

If a sequencer is unable to make a base call with sufficient confidence then it will normally substitute an N rather than a conventional base call. This module plots out the percentage of base calls at each position for which an N was called.



It's not unusual to see a very low proportion of Ns appearing in a sequence, especially nearer the end of a sequence. However, if this proportion rises above a few percent it suggests that the analysis pipeline was unable to interpret the data well enough to make valid base calls.

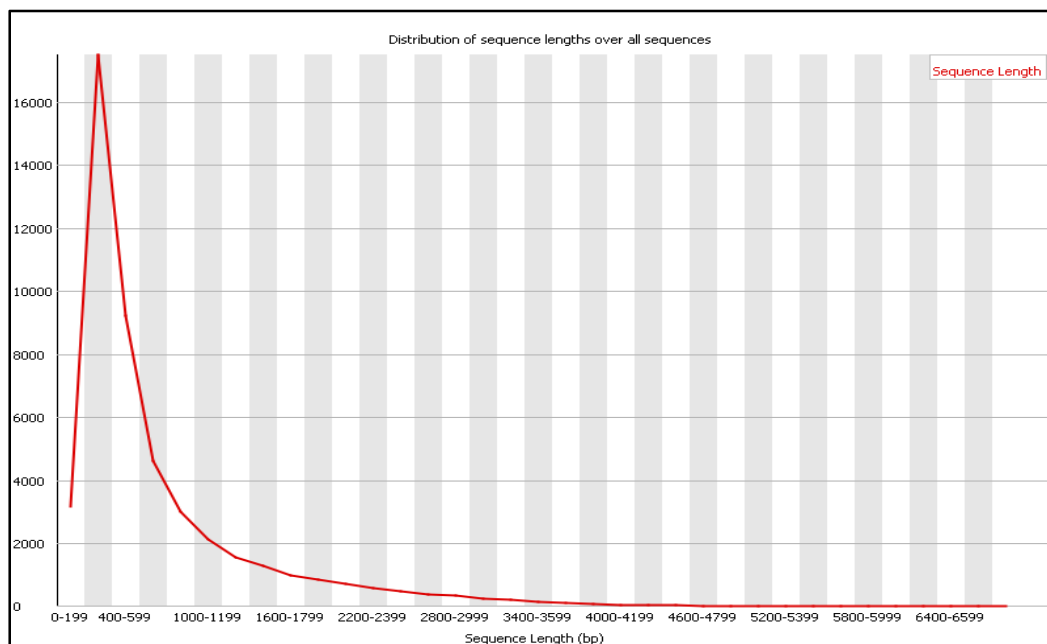
This module raises a warning if any position shows an N content of >5%. This module will raise an error if any position shows an N content of >20%.

(viii) Sequence Length Distribution

Some high throughput sequencers generate sequence fragments of uniform length, but others can contain reads of wildly varying lengths. Even within uniform length libraries some pipelines will trim sequences to remove poor quality base calls from the end.

This module generates a graph showing the distribution of fragment sizes in the file which was analyzed. In many cases this will produce a simple graph showing a peak only at one size, but for variable length FastQ files this will show the relative amounts of each different size of sequence fragment.

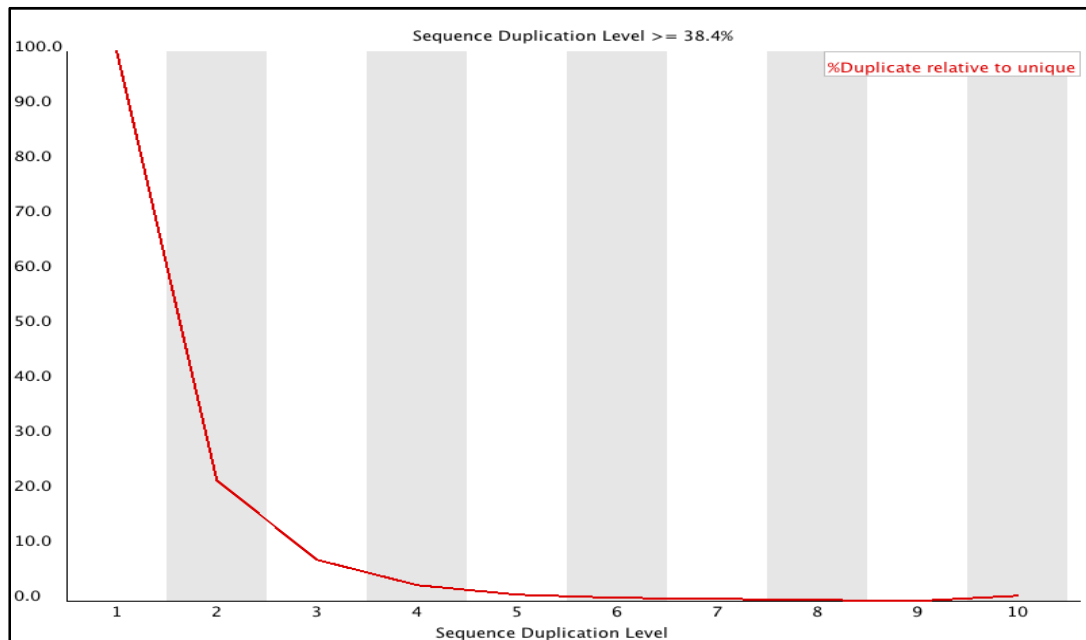
This module will raise a warning if all sequences are not the same length. This module will raise an error if any of the sequences have zero length.



(ix) Duplicate Sequences

In a diverse library most sequences will occur only once in the final set. A low level of duplication may indicate a very high level of coverage of the target sequence, but a high level of duplication is more likely to indicate some kind of enrichment bias (eg PCR over amplification).

This module counts the degree of duplication for every sequence in the set and creates a plot showing the relative number of sequences with different degrees of duplication.



To cut down on the memory requirements for this module only sequences which occur in the first 200,000 sequences in each file are analyzed, but this should be enough to get a good impression for the duplication levels in the whole file. Each sequence is tracked to the end of the file to give a representative count of the overall duplication level. To cut down on the amount of information in the final plot any sequences with more than 10 duplicates are placed into the 10 duplicates category - so it's not unusual to see a small rise in this final category. If you see a big rise in this final category then it means you have a large number of sequences with very high levels of duplication.

Because the duplication detection requires an exact sequence match over the whole length of the sequence any reads over 75bp in length are truncated to 50bp for the purposes of this analysis. Even so, longer reads are more likely to contain sequencing errors which will artificially increase the observed diversity and will tend to under represent highly duplicated sequences.

This module will issue a warning if non-unique sequences make up more than 20% of the total. This module will issue an error if non-unique sequences make up more than 50% of the total.

(x) Overrepresented Sequences

A normal high-throughput library will contain a diverse set of sequences, with no individual sequence making up a tiny fraction of the whole. Finding that a single sequence is very overrepresented in the set either means that it is highly biologically significant, or indicates that the library is contaminated, or not as diverse as you expected.

This module lists all of the sequence which makes up more than 0.1% of the total. To conserve memory only sequences which appear in the first 200,000 sequences are tracked to the end of the file. It is therefore possible that a sequence which is overrepresented but doesn't appear at the start of the file for some reason could be missed by this module.

For each overrepresented sequence the program will look for matches in a database of common contaminants and will report the best hit it finds. Hits must be at least 20bp in length and have no more than 1 mismatch. Finding a hit doesn't necessarily mean that this is the source of the contamination, but may point you in the right direction. It's also worth pointing out that many adapter sequences are very similar to each other so you may get a hit reported which isn't technically correct, but which has very similar sequence to the actual match.

Because the duplication detection requires an exact sequence match over the whole length of the sequence any reads over 75bp in length are truncated to 50bp for the purposes of this analysis. Even so, longer reads are more likely to contain sequencing errors which will artificially increase the observed diversity and will tend to under represent highly duplicated sequences.

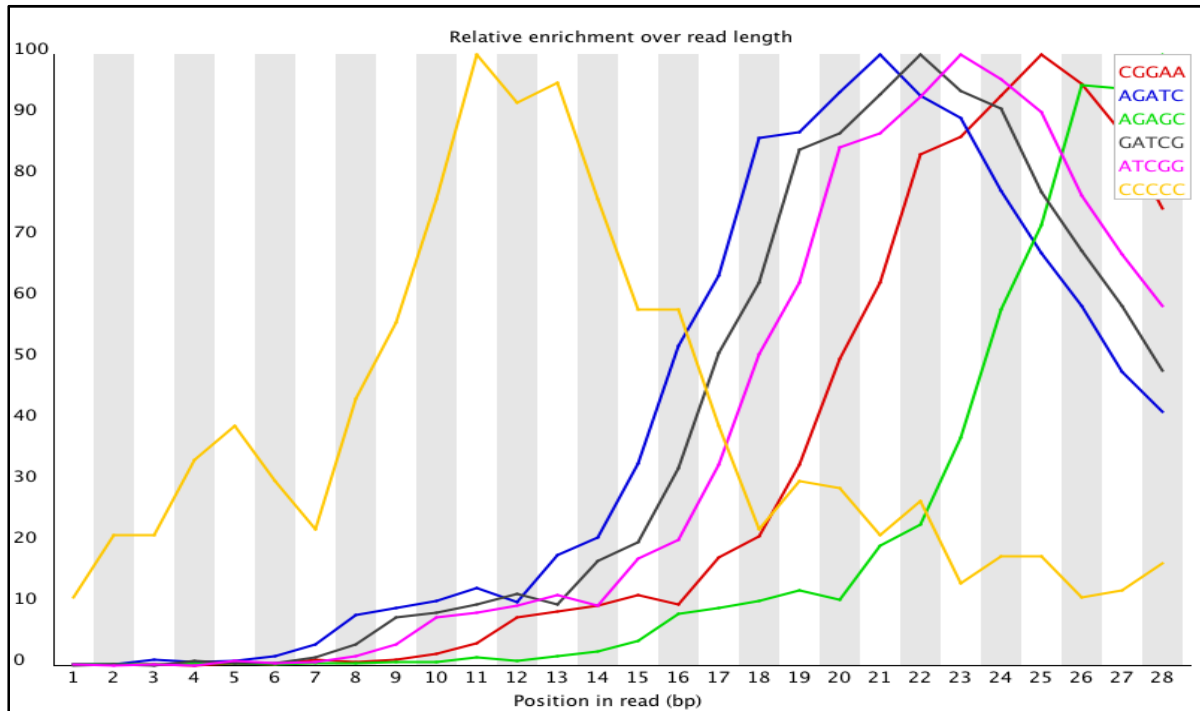
This module will issue a warning if any sequence is found to represent more than 0.1% of the total. This module will issue an error if any sequence is found to represent more than 1% of the total.

(xi) Over represented Kmers

The analysis of overrepresented sequences will spot an increase in any exactly duplicated sequences, but there is a different subset of problems where it will not work.

- If you have very long sequences with poor sequence quality then random sequencing errors will dramatically reduce the counts for exactly duplicated sequences.
- If you have a partial sequence which is appearing at a variety of places within your sequence then this won't be seen either by the per base content plot or the duplicate sequence analysis.

This module counts the enrichment of every 5-mer within the sequence library. It calculates an expected level at which this k-mer should have been seen based on the base content of the library as a whole and then uses the actual count to calculate an observed/expected ratio for that k-mer. In addition to reporting a list of hits it will draw a graph for the top 6 hits to show the pattern of enrichment of that Kmer across the length of your reads. This will show if you have a general enrichment, or if there is a pattern of bias at different points over your read length.



Any k-mer showing more than a 3 fold overall enrichment or a 5 fold enrichment at any given base position will be reported by this module.

To allow this module to run in a reasonable time only 20% of the whole library is analyzed and the results are extrapolated to the rest of the library.

This module will issue a warning if any k-mer is enriched more than 3 fold overall, or more than 5 fold at any individual position. This module will issue an error if any k-mer is enriched more than 10 fold at any individual base position.

2.6 Practical

2.6.1 QC analysis using Script

Run the script using terminal, within the folder containing whole genome sequencing read data, and analyze the result

2.6.2 QC analysis using Tool: FastQC

Load the whole genome sequence read data to the software to analyze and save results.

3. Genome Alignment & Assembly

In bioinformatics, **sequence assembly** refers to aligning and merging fragments of a much longer DNA sequence in order to reconstruct the original order of the sequence. Genome assembly is simply putting sequenced fragments of DNA into their correct chromosomal positions. This is needed as DNA sequencing technology cannot read whole genomes in one go, but rather reads small pieces of between 20 and 1000 bases, depending on the technology used. Typically the short fragments, called reads, result from sequencing genomic DNA, or gene transcript (ESTs).

3.1 Genome Assemblers

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. As the sequenced organisms grew in size and complexity (from small viruses over plasmids to bacteria and finally eukaryotes), the assembly programs used in these genome projects needed to increasingly employ more and more 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;
- Errors in the fragments from the sequencing instruments, which can confound assembly.

3.1.1 De-novo vs. mapping (reference) assembly

In sequence assembly, two different types can be distinguished:

- **De-novo:** assembling short reads to create full-length (sometimes novel) sequences. Overlapping reads are presumed to be from the same area of the genome, for *de-novo* assembly, read depth should be more.
- **Mapping or Reference:** assembling reads against an existing backbone sequence taken as reference, building a sequence that is similar but not necessarily identical to the backbone sequence. Combines all sequencing reads into contigs based on sequence similarity between reads and reference.

3.1.2 Assessing Assembly Quality

Common measures of quality are:

- number and sizes of contigs

- Assumption: few large contigs is better than many small contigs.
- True because there are fewer gaps in the former, but, does not account for the possibility of misassemblies.

3.1.3 Tools/ Software's for Assembly

- TIGR Assembler
- Velvet (Denovo)
- Maq (Reference)
- Reference assembly& Alignment using BWA tool and Visualization of alignment using SAM

3.1.4 Applications of Genome assembly

- Generating and interpreting alignment status and reports
- Genome variation calling (finding SNP's, *indels*)
- Variation annotation and Viewing

3.2 Practical

2.2.1 Reference assembly- BWA

2.2.2 De-novo Assembly - Velvet

3.2.1 Reference Assembly

(i) Genome Alignment & Assembly with reference using BWA tool

BWA is a fast light-weighted tool that aligns relatively short sequences (queries) to a sequence database (target), such as the human reference genome. It implements two different algorithms, both based on Burrows-Wheeler Transform (BWT). The first algorithm is designed for short queries up to ~200bp with low error rate (<3%). It does gapped global alignment w.r.t. queries, supports paired-end reads, and is one of the fastest short read alignment algorithms to date while also visiting suboptimal hits. The second algorithm, BWA-SW, is designed for long reads with more errors. It performs heuristic Smith-Waterman-like alignment to find high-scoring local hits (and thus chimera). On low-error short queries, BWA-SW is slower and less accurate than the first algorithm, but on long queries, it is better.

For both algorithms, the database file in the FASTA format must be first indexed with the '**index**' command, which typically takes a few hours. The first algorithm is implemented via the

'aln' command, which finds the suffix array (SA) coordinates of good hits of each individual read, and the **'samse/sampe'** command, which converts SA coordinates to chromosomal coordinate and pairs reads (for 'sampe'). The second algorithm is invoked by the **'bwasw'** command. It works for single-end reads only.

After quality check we have to download reference sequence (eg : NC_000962.fna or fasta) and index it as our reference for aligning the reads, and mtb in the first command is the database prefix name and that should be the same in remaining steps. After indexing align read 1 and read 2 separately with indexed reference. Finally merge the aligned reads 1 and 2 with reference to remove redundancy.

Visualization in the 5th step will allow us to read output data using SAM tools(Sequence Alignment/Map), From sam file we can obtain information like, how much genome is covered with reference to reference genome, if the read is not aligning to the reference, then * is placed in 3rd column of sam file.

1. Indexing our sequence with reference.fna for bwa alignment

```
~/Programs/bwa-0.5.9/bwa index -p mtb NC_000962.fna
```

2. Aligning filtered read1 (bwa_1.aln is output and ERR015582_1_filtered_matched.fastq is input, mtb is the database)

```
~/Programs/bwa-0.5.9/bwaaln -t 2 -e 15 -I -f bwa_1.aln mtb  
ERR015582_1_filtered_matched.fastq
```

3. Aligning filtered read2

```
~/Programs/bwa-0.5.9/bwaaln -t 2 -e 15 -I -f bwa_2.aln mtb  
ERR015582_2_filtered_matched.fastq
```

4. Merging aligned reads1&2 with reference (aln.sam is the output file, SAM is tool used for visualization)

```
~/Programs/bwa-0.5.9/bwasampe -f aln.sammtn bwa_1.aln bwa_2.aln  
ERR015582_1_filtered_matched.fastq ERR015582_2_filtered_matched.fastq
```

5. Visualisation using SAM

```
headaln.sam
```

(ii) Alignment Status and Reports

1. Writing unaligned and aligned reads in alignment statistics (input is aln.sam and output is Alignment_statistics.txt)


```
awk '{if($3!="*"){i++} else {print $0> "unaligned.sam";j++}END{percReads=i/j*100;print
"Total Reads\t" j "\nReads aligned\t" i "\n% Reads Aligned\t" percReads}'
aln.sam>Alignment_Statistics.txt
```

2. To view sam file (output aln.bam)

```
~/Programs/samtools-0.1.17/samtools view -o aln.bam -b -S -T NC_000962.fna aln.sam
```

3. Sorting bam file (input aln.bam, output aln_sorted)

```
~/Programs/samtools-0.1.17/samtools sort aln.bamaln_sorted
```

4. Generating pileup format from BAM file (to identify no of reads aligned or occurring at one base pair position to test quality of each bases, 4th column in mpileup is the no of mappings to reference for each base)

```
~/Programs/samtools-0.1.17/samtoolsmpileup -6 -s -f NC_000962.fna aln_sorted.bam
```

```
>aln_mpileup.txt
```

5. Alignment statistics (will give % coverage, average depth, X coverage etc. BAM and SAM file contains all information regarding variation, read depth etc. from which we are writing it in to the mpileup.txt)

```
awk
'{totalRD=totalRD+$4;i++}END{percCov=i/64022747*100;avgRD=totalRD/64022747;print"Total Genome Size\t64022747\nGenome Covered\t"i"\n% Coverage\t" percCov "\nAvg Read Depth\t" avgRD}' aln_mpileup.txt >>Alignment_Statistics.txt
```

6. X average coverage

```
awk
'{if($4>=1){i++;if($4>=5){i++;if($4>=10){k++;if($4>=15){l++;if($4>=20){m++;if($4>=30){n++;if($4>=40){o++}}}}}}};all++}END
{percI=i/4639675*100;percJ=j/4639675*100;percK=k/4639675*100;percL=l/4639675*100;percM=m/4639675*100;percN=n/4639675*100; percO=o/4639675*100;print "% Coverage at 1X\t"
percI "\n% Coverage at 5X\t" percJ "\n% Coverage at 10X\t" percK "\n% Coverage at 15X\t"
percL "\n% Coverage at 20X\t" percM "\n% Coverage at 30X\t" percN "\n% Coverage at 40X\t"
percO}' aln_mpileup.txt >>Alignment_Statistics.txts
```

3.2.2 De-novo Assembly

Velvet is a novel set of de Bruijn graph-based sequence assembly methods for very short reads that can both remove errors and, in the presence of read pair information, resolve a large number of repeats. With unpaired reads, the assembly is broken when there is a repeat longer than the k -

mer length. With the addition of short reads in read pair format, many of these repeats can be resolved, leading to assemblies similar to draft status in bacteria and reasonably long (~5 kb) SCSCs in eukaryotic genomes. Velvet can convert high-coverage very short reads into reasonably sized contigs with no additional information. With additional paired read information to resolve small repeats, almost complete genomes can be assembled. We believe the Velvet framework will provide a rich set of different algorithmic options tailored to different tasks and thus provide a platform for cheap de novo sequence assemblies, eventually for all genomes.

(i) Steps involved in *De-novo* Assembly (Velvet)

1. Shuffle Paired end reads

```
~/Programs/velvet_1.1.05/shuffleSequences_fastq.pl ERR015582_1_filtered_matched.fastq  
ERR015582_2_filtered_matched.fastq ERR015582_shuffled_filtered_matched.fastq
```

2. calculating number of reads (to enter as input for calculating Kmer, only one filtered shuffled we have to do. because both contain same data)

```
grep "@ERR" ERR015582_1_filtered_matched.fastq -c
```

3. Calculate K-Mer

```
perl ~/Programs/calculate_Kmer.pl
```

4. Velvet for graph generation for kmer range from 31-49 with step 2 (hassing or indexing means indexing reads with contigs and keeping that doing assembly)(assembly optimisation1)

```
~/Programs/velvet_1.1.05/velvethmultipleKmer 31,49,2 -fastq -shortPaired  
ERR015582_shuffled_filtered_matched.fastq
```

5. velvetg to assemble and generate contigs for Kmer31 (assembly command to get assembly status) (optimisation 2)

```
~/Programs/velvet_1.1.05/velvetg multipleKmer_31/ -ins_length_sd 20 -ins_length 200 -  
read_trkg yes -min_contig_lgth 100 -cov_cutoff auto -exp_cov auto -scaffolding yes -  
unused_reads yes -alignments yes
```

6. Velvetg to assemble and generate contigs for kmer33

```
~/Programs/velvet_1.1.05/velvetg multipleKmer_33/ -ins_length_sd 20 -ins_length 200 -  
read_trkg yes -min_contig_lgth 100 -cov_cutoff auto -exp_cov auto -scaffolding yes -  
unused_reads yes -alignments yes
```

7. Velvetg to assemble and generate contigs for kmer35

```
~/Programs/velvet_1.1.05/velvetg multipleKmer_35/ -ins_length_sd 20 -ins_length 200 -  
read_trkg yes -min_contig_lgth 100 -cov_cutoff auto -exp_cov auto -scaffolding yes -  
unused_reads yes -alignments yes
```

8. Velvetg to assemble and generate contigs for kmer37

```
~/Programs/velvet_1.1.05/velvetg multipleKmer_37/ -ins_length_sd 20 -ins_length 200 -  
read_trkg yes -min_contig_lgth 100 -cov_cutoff auto -exp_cov auto -scaffolding yes -  
unused_reads yes -alignments yes
```

9. Velvetg to assemble and generate contigs for kmer39

```
~/Programs/velvet_1.1.05/velvetg multipleKmer_39/ -ins_length_sd 20 -ins_length 200 -  
read_trkg yes -min_contig_lgth 100 -cov_cutoff auto -exp_cov auto -scaffolding yes -  
unused_reads yes -alignments yes
```

10. Velvetg to assemble and generate contigs for kmer41

```
~/Programs/velvet_1.1.05/velvetg multipleKmer_41/ -ins_length_sd 20 -ins_length 200 -  
read_trkg yes -min_contig_lgth 100 -cov_cutoff auto -exp_cov auto -scaffolding yes -  
unused_reads yes -alignments yes
```

11. Velvetg to assemble and generate contigs for kmer43

```
~/Programs/velvet_1.1.05/velvetg multipleKmer_43/ -ins_length_sd 20 -ins_length 200 -  
read_trkg yes -min_contig_lgth 100 -cov_cutoff auto -exp_cov auto -scaffolding yes -  
unused_reads yes -alignments yes
```

12. Velvetg to assemble and generate contigs for kmer45

```
~/Programs/velvet_1.1.05/velvetg multipleKmer_45/ -ins_length_sd 20 -ins_length 200 -  
read_trkg yes -min_contig_lgth 100 -cov_cutoff auto -exp_cov auto -scaffolding yes -  
unused_reads yes -alignments yes
```

13. Velvetg to assemble and generate contigs for kmer47

```
~/Programs/velvet_1.1.05/velvetg multipleKmer_47/ -ins_length_sd 20 -ins_length 200 -  
read_trkg yes -min_contig_lgth 100 -cov_cutoff auto -exp_cov auto -scaffolding yes -  
unused_reads yes -alignments yes
```

14. Generate assembled reads stats:

```
tail Log -n 1 | awk -F "/" '{asm=$1;tot=$2;sub(/.*using
/, "", asm);sub(/reads/, "", tot);percAsm=asm/tot*100;print
"AssembledReads\t"asm"\nTotalReads\t"tot"\nPercentageAssembled\t"percAsm}' >
assembly_stats.txt
```

15. Generate sorted contig length information | Max

```
awk
'BEGIN{min=100000000;}{if($1~/>/){i++;if(seq!=""){len=length(seq);tLen=tLen+len;print
header"\t"len;seq="";if(max<len){max=len};if(min>len){min=len}};header=$0}else{seq=seq
""$0}}END{len=length(seq);tLen=tLen+len;aLen=tLen/i;printheadert"\t"len;if(max<len){max
=len};if(min>len){min=len};print "Total No. of Contigs\t"i"\nTotal Assembly
Length\t"tLen"\nMaxContig Length\t"max"\nMinContig Length\t"min"\nAverage sequence
Length\t"aLen}>> "assembly_stats.txt"}' contigs.fa | sort -k 2,2nr >contig_lengths.txt
```

16. Get n50 count (last optimisation)

```
tail Log -n 1 | awk -F "\t" '{nffty=$1;sub(/.*n50 of /, "", nffty);sub(/, max.*/, "", nffty);print "n50\t"
nffty}' >> assembly_stats.txt
```

(ii) Genome variation calling (finding SNP's, indels)

Next-generation sequencing technologies have been widely used for effective, easy and in-depth investigation of genetic variation, including SNPs and InDels (insertion/deletions), to a better understanding of the organism.

BAM and SAM file contains all information on variation, read depth etc. from which we are writing it in to the mpileup.txt for variation calling and annotation. From mpileup.txt alignment file and BAM file gaps and variations are to be identified for variation annotation. Variation calculation is done by from mpileup and sorted BAM file, from which it is converted to Binary format variation and then to text format variation.

1. Variation calling (generating BCF file from BAM- BCF- Binary variation file)

```
~/Programs/samtools-0.1.17/samtoolsmpileup -ugf NC_000962.fna aln_sorted.bam |
~/Programs/samtools-0.1.17/bcftools/bcftools view -bvcg ->var.raw.bcf
```

2. Converting BCF (binary variation file to VCF (text variation file)

```
~/Programs/samtools-0.1.17/bcftools/bcftools view var.raw.bcf>var.raw.vcf
```

(iii) Variation annotation

Download reference ptt or Gff or gtf from go to ftp://ftp.ncbi.nih.gov/genomes/Bacteria/Mycobacterium_tuberculosis_H37Rv_uid57777/ and edit the ptt file and save it as reference name.ptt (eg.NC_000962.ptt). Then open .vcf file and delete everything till (chroms) and save it as (SNP_table.txt).

1. Map SNP's to PTT by reference name

```
join -t $'\t' -1 1 -2 1 SNP_table.txt NC_000962.ptt > joined_SNP_annotation.txt
```

2. Filter SNP's falling in genomic elements

```
awk '{if($2>=$11 && $2<=$12) {print $0}}' joined_SNP_annotation.txt > filtered_SNP_annotation.txt
```

3. Indexing sorted Bam file

```
~/Programs/samtools-0.1.17/samtools index aln_sorted.bam
```

4. Running SAM tools viewer

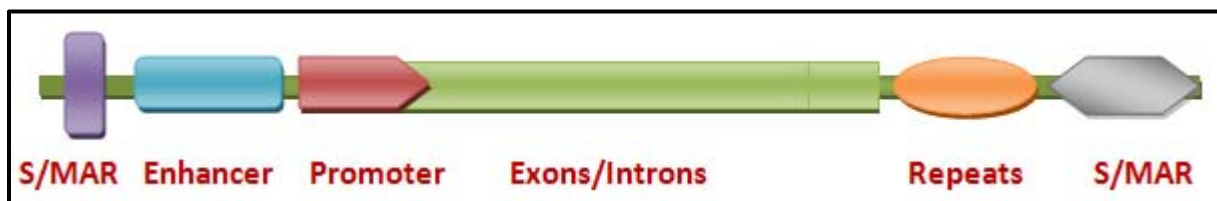
```
~/Programs/samtools-0.1.17/samtoolstviewaln_sorted.bam NC_000962.fna
```


4. Genome Annotation

Obtaining the sequence from whole genome sequencing and assembling is not the end of a genome project, however finding and attaching the structural elements and its related function are the next major steps, which are called “**Genome Annotation**”. The process of identifying the locations of genes and all of the coding regions in a genome and determining what those genes do. An annotation (irrespective of the context) is a note added by way of explanation or commentary. Once a genome is sequenced, it needs to be annotated to make sense of it. Annotation is the process of adding pertinent information about the raw DNA sequences to the genome or process of attaching biological information to DNA or Protein sequences by describing different regions of the code and identifying which regions can be called genes and thereby its products and functions. This include spotting locations of genes, total number of genes, coding regions, intron-exon structure, start and stop codons, intron lengths, alternative splicing, SNP's, InDels and untranslated regions (UTRs) as well as and determining what those genes do along with the gene product and functional information. Once a genome is sequenced, it needs to be annotated to make sense of it. With many genomes now sequenced, computational annotation methods to characterize genes and proteins from their sequence are increasingly important. It consists of two main steps:

- ❖ Identifying elements on the genome- **gene structure prediction**
- ❖ Attaching biological information to these elements- **gene function prediction**

There are various parts within the gene with different functions, some may code for protein, others may contain regulatory information, some may form introns and will not be translated and their function is still unclear. The diagram shown below represents fragment of DNA, with single hypothetical gene. Each region has to be annotated from DNA sequences based on similarity searches or literature reviews.



Obviously computer programs are essential to this process; however, human brains are often required to evaluate computer-generated gene models. Several Automatic annotation tools are available that are highly accurate. Annotation tools can perform all this by computer analysis, as opposed to manual annotation which involves human expertise. Ideally, these approaches co-exist and complement each other in the same annotation pipeline.

The basic level of annotation uses BLAST for finding similarities, and then annotating genomes based on that. However, much additional information is available to annotation platform nowadays. Some databases use genome context information, similarity scores, experimental data, and integrations of other resources to provide genome annotations through their Subsystems approach. Other databases such as Ensembl rely on both curated data sources as well as a range of different software tools in their automated genome annotation pipeline.

4.1 Annotation Methods

- ❖ *Ab-Initio* - Statistics-based methods
- ❖ **Homology** based methods

The first hurdle for any functional annotation process is to define 'function'. In general however, the problem is multi-dimensional: a protein can have a molecular function, a cellular role, and be part of a functional complex or pathway (these are the distinctions used in the Gene Ontology). Furthermore, certain aspects of molecular function can be illustrated by multiple descriptive levels (for example, the coarse 'enzyme' category versus a more specific 'protease' assignment). Even the more detailed definition would not reveal the cellular role of the protein (apoptosis, metabolism, blood coagulation, and so on). Most function-prediction methods, both sequence and structure based, rely on inferring relationships between proteins that permit the transfer of functional annotations and binding specificities from one to the other. A notable challenge here is deciphering the connection between the detected similarities (structural or in sequence) and the actual level of functional relatedness. Function is often associated with domains, and another problem is the identification of functional domains from sequence alone. The accuracy of current methods for predicting domain boundaries is not yet completely satisfactory. Several methods provide reliable predictions if a structural template for the protein is available, but when this is not the case, one is left with the problem of whether the experimental annotation used for the inference refers to the same domain for which the sequence similarity/motif is established.

The function of a protein can also be inferred from its evolutionary relationship with proteins of known function, provided that the relationship is properly inspected. Orthologous proteins in different species most often share function, but paralogy (that is, divergence following duplication of the original gene) does not guarantee common function. Distinguishing between orthology and paralogy can be attempted on the basis of observed sequence-similarity patterns, by analyzing the specific conservation pattern of residues responsible for function in the family, or on the basis of the protein structure (either experimentally determined or modeled). In all cases, this requires the clustering of proteins into evolutionary families, which can be achieved using similarity-detection tools such as BLAST or profiling tools based on multiple sequence alignments, for example, PSI-BLAST. Several available resources provide pre-compiled family assignments for proteins on a genomic scale, based only on their sequence. Resources can be subdivided into those that consider full-length sequences and those based on domains or motifs

that map to certain sub-sequences. In both cases, the degree of granularity of the classification is important, as this is related to the level of functional features that a group of proteins is expected to share.

The accuracy of sequence-based methods is affected by the type and amount of information on the specific protein family but, overall, they seem to be reasonably accurate. Their success rate has been shown to be greater than 70% when tested on a limited dataset (all structures solved by the Midwest Center for Structural Genomics during the first five years of the Protein Structure Initiative)

4.2 Structural & Functional Genome Annotation

Structural annotation consists of the identification of genomic elements.

- Open reading frame and their localization
- Gene structure
- Coding regions
- Location of regulatory motifs

Functional annotation consists of attaching biological information to genomic elements.

- Biochemical function
- Biological function
- Involved regulation and interactions
- Expression

These steps may involve both biological experiments and *in-silico* analysis. A variety of software tools have been developed to permit scientists to view and share genome annotations.

- Structural annotation for prokaryotes & eukaryotes – PRODIGAL & AUGUSTUS
- Functional annotation (based gene level homology and protein level homology)

4.3 Practical

4.3.1 STRUCTURAL ANNOTATION

Eukaryotes- AUGUSTUS

Structural annotation for eukaryotes - Augustus

Gene structural prediction with a genome model

```
~/Programs/augustus.2.5.5/bin/augustus --strand=both --genemodel=partial --singlestrand=true --  
alternatives-from-evidence=true --alternatives-from-sampling=true --progress=true --gff3=on --  
uniqueGeneId=true --species=magnaporthe_griseaPhyca.fasta>out.gff
```

Prokaryotes - PRODIGAL

Structural annotation for prokaryotes -Prodigal

```
~/Programs/prodigal.v2_60.linux -a _mtb_prot_file.fa -g 11 -d mtb_gene_seq.fa -f gff -i  
contigs.fa -o genes_quality.txt -s genes_score.txt -t mtb_training_file.txt
```

4.3.2 FUNCTIONAL ANNOTATION

Steps for functional annotation

Functional annotation requires creating database for homology searches. First download complete swissprot database and Complete NCBI and setup database for homology search using the command.

Below each command run separately for nucleotide and then for protein

1. setup blast database for nucleotide

```
formatdb -i ncbi.fasta -p F -o F -n database1
```

2. setup blast database for protein

```
formatdb -i uniprot_sprot.fasta -p T -o F -n database2
```

3. Blasting the annotations

(i) For nucleotide

```
blastall -p blastn -i sequence.fasta -d database1 -a 2 -m 8 -o blast_result_nucleotide.txt
```

(ii) For Protein

```
blastall -p blastx -i contigs.fa -d database2 -a 2 -m 8 -o blast_result_protein.txt
```

4. sorting for blast min E-value

(i) For nucleotide

```
sort -k 1,1 -k 11,11g blast_result.txt > sorted_blast_result_nucleotide.txt
```

(ii) For Protein

```
sort -k 1,1 -k 11,11g blast_resul_prot.txt > sorted_blast_result_protein.txt
```

5. Further filtering for best blast hit

(i) For nucleotide

```
awk '{if($1!=id){print};id=$1}' sorted_blast_result_nucleotide.txt>best_blast_result1.txt
```

(ii) For Protein

```
awk '{if($1!=id){print};id=$1}' sorted_blast_result_protein.txt>best_blast_result2.txt
```

6. Removing Positive strand blast hits

```
awk '{if($9>$10){print}}' best_blast_result1.txt> blast_result_neg.txt
```

7. Removing negative strand blast hits

```
awk '{if($10>$9){print}}' best_blast_result1.txt > blast result_pos.txt
```

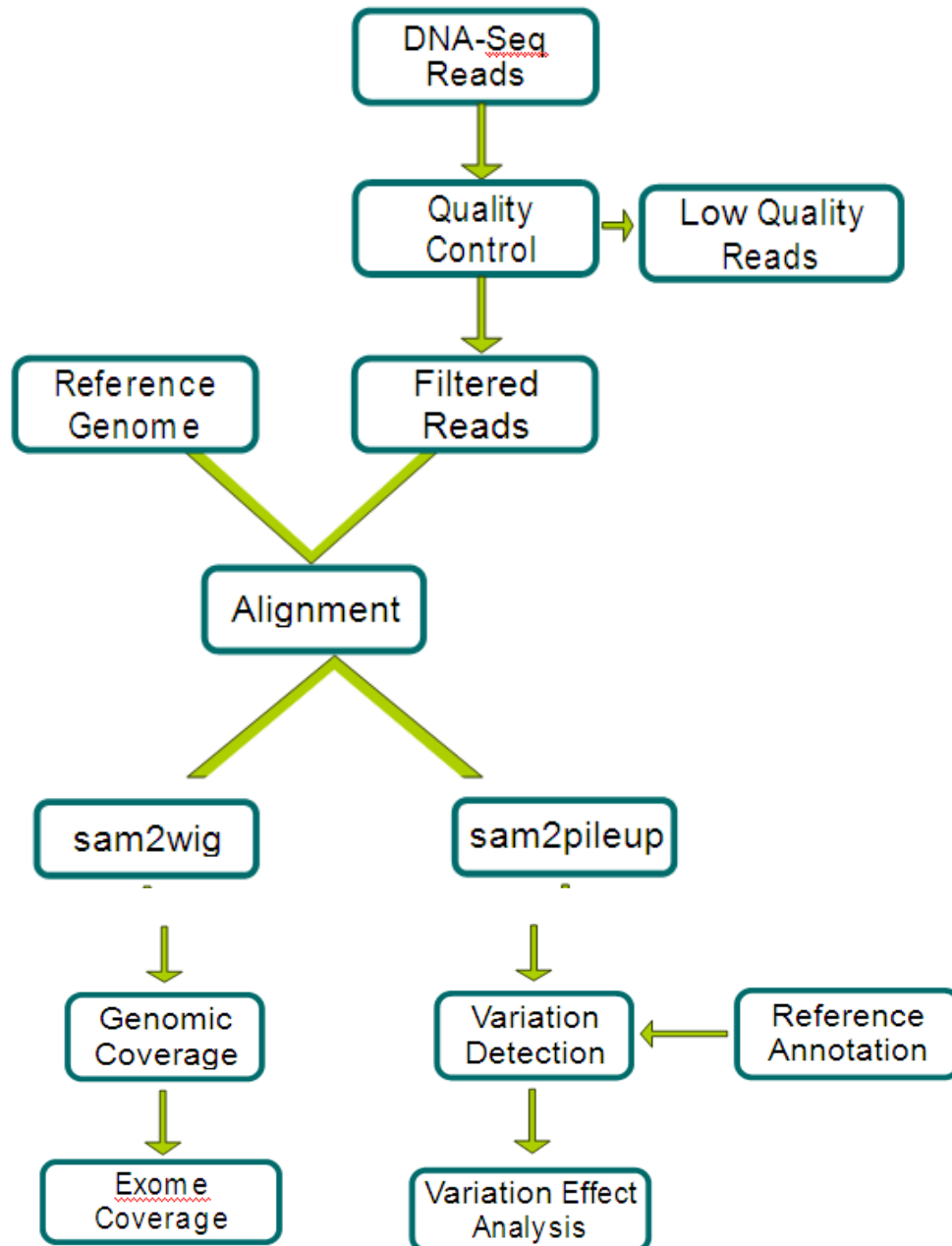
4.4 Further Reading

1. Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT, Harris MA, Hill DP, Issel-Tarver L, Kasarskis A, Lewis S, Matese JC, Richardson JE, Ringwald M, Rubin GM, Sherlock G: Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat Genet* 2000, 25:25-29.
2. Tress M, Cheng J, Baldi P, Joo K, Lee J, Seo JH, Lee J, Baker D, Chivian D, Kim D, Ezkurdia I: Assessment of predictions submitted for the CASP7 domain prediction category. *Proteins* 2007, 69(Suppl 8):137-151.
3. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ: Basic local alignment search tool. *J MolBiol* 1990, 215:403-410.
4. Altschul SF, Madden TL, Schäffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ: Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res* 1997, 25:3389-3402.
5. Watson JD, Sanderson S, Ezersky A, Savchenko A, Edwards A, Orengo C, Joachimiak A, Laskowski RA, Thornton JM: Towards fully automated structure-based function prediction in structural genomics: a case study. *J MolBiol* 2007, 367:1511-1522.

5. Whole Genome Re-sequencing & NGS Data Analysis

Rohit Nandan Shukla , Bionivid Technology Private Limited

5.1 Analysis Work flow



5.2 Sequence Reads QC & Filtering

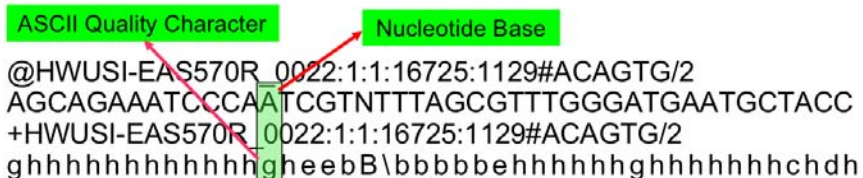
NGS platforms have their own share of quality issues which can be significant from lab- to-lab, batch-to-batch and even within chip/slide variations.

Sample Raw Data:

```
@HWUSI-EAS570R_0022:1:1:2170:1105#ACAGTG/2
GCGCGATAANNNGATNNNNNNNNNNNTACNNNNNCANACCNNNNNNACATCGC
+HWUSI-EAS570R_0022:1:1:2170:1105#ACAGTG/2
hhhhf`abaBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
@HWUSI-EAS570R_0022:1:1:4096:1100#ACAGTG/2
CGCACCTTTNNNTTCNNNNNNNNNNNTTANNNNNCANNTCNNNNNNNATATCG
+HWUSI-EAS570R_0022:1:1:4096:1100#ACAGTG/2
hhhhhcbdaBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
@HWUSI-EAS570R_0022:1:1:11046:1102#ACAGTG/2
AGTGC GTTCNNNCATNNNNNNNNNNNACGNNNNNAANTGCNNNNNNNTTTAAT
+HWUSI-EAS570R_0022:1:1:11046:1102#ACAGTG/2
hhghhdddddBBB^BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
@HWUSI-EAS570R_0022:1:1:12429:1105#ACAGTG/2
ACCCCAAGCCNCCCNNNNNNNNNNTCTCNNNTGTTTTNNNNNNGTGATCC
+HWUSI-EAS570R_0022:1:1:12429:1105#ACAGTG/2
hhhhhfffffBBaaaBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
```

5.2.1 QC & Filtering Parameters:

- ❖ Low complexity
- ❖ Low quality reads filter
- ❖ Reads with high N %
- ❖ Adapter contamination



The diagram illustrates the conversion of a nucleotide base to an ASCII quality character. A green box labeled 'ASCII Quality Character' points to the character 'g' in the sequence. Another green box labeled 'Nucleotide Base' points to the 'g' in the sequence. The sequence shown is:

```
@HWUSI-EAS570R_0022:1:1:16725:1129#ACAGTG/2
AGCAGAAATCCCAATCGTNTTTAGCGTTTGGGATGAATGCTACC
+HWUSI-EAS570R_0022:1:1:16725:1129#ACAGTG/2
ghhhhhhhhhhhhhgheebB\bbbbbehhhhhghhhhhhhchdh
```

ASCII Value: g = 103
 Quality Score = ASCII Value – 64
 = 103 – 64
 = 39

5.3 Mapping to Reference

5.3.1 Indexing FASTA Reference**Usage:** bowtie-build [options]* <reference_in> <ebwt_outfile_base>

reference_in	comma-separated list of files with ref sequences
ebwt_outfile_base	write Ebwt data to files with this dir/basename

Options:**-f** reference files are Fasta (default)**Command:***bowtie-build* TAIR10_chr_all.fasta TAIR10**Reference****Alignment: Bowtie:****Usage:** bowtie [options]* <ebwt> {-1 <m1> -2 <m2> | --12 <r> | <s>}
[<hit>]**Options:**

--solexa1.3-quals	input quals are from GA Pipeline ver. >= 1.3
-t/--time	print wall-clock time taken by search phases
--al <fname>	write aligned reads/pairs to file(s) <fname>
--un <fname>	write unaligned reads/pairs to file(s) <fname>
-p/--threads <int>	number of alignment threads to launch (default:

1)

--chunkmbs <int>	max megabytes of RAM for best-first search
frames (def: 64)	

Command:

bowtie-S-solexa1.3-quals -t-alaligned.fq -un

unaligned.fq -p4 -chunkmbs 100TAIR10 s_4_sequence.txt s_4_sequence.txt.sam

5.4 Alignment Visualization

Tablet Visualization tool

Open Assembly

Select assembly files:

Primary assembly file or URL:

Reference/consensus file or URL:

Current status: Assembly - Unknown

Notes:

- Tablet currently supports ACE, AFG, MAQ (text), SOAP, SAM, and BAM assemblies.
- Reference files (if needed for MAQ, SOAP, and SAM) can be in FASTA or FASTQ format.
- BAM files need to have been indexed, and a FASTA reference file must be included too.
- [Unsure how to get started? Click here to open an example assembly.](#)



5.5 Variation Detection & Visualization

5.5.1 Generate variations pileup file from sam (alignment) file

SAM to BAM conversion

```
samtools view -bS -o sequence.bam sequence.sam
```

Sorting BAM file

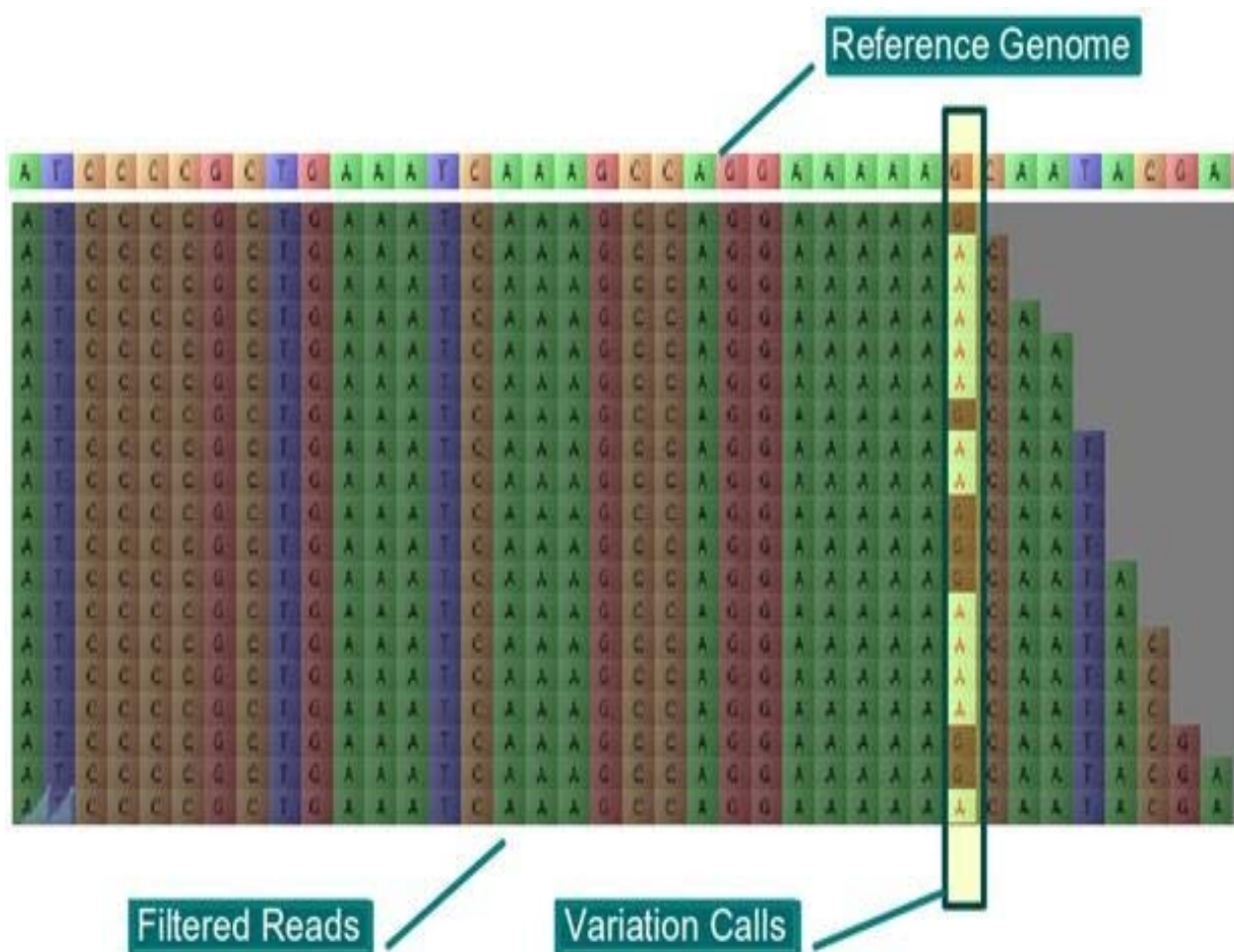
```
samtools sort sequence.bam sequence.sorted
```

Generate consensus pileup from BAM file

```
samtools pileup -cv -f TAIR10_chr_all.fasta sequence.sorted.bam >sequence.pileup
```

Filter variations from pileup

```
samtools.pl varFilter sequence.pileup | awk '$6>=20 && $8>=20' > sequence.final.pileup
```



5.5.2 Variation Table

Reference	Position	Ref base	Consensus Base	Quality
Chr1	1	G	A	4
Chr1	29	C	T	15
Chr1	69	C	A	22
Chr2	5	A	T	4
Chr2	28	C	G	15
Chr2	53	T	G	22
Chr2	63	A	G	22
Chr3	5	T	C	7
Chr3	33	G	A	11
Chr3	62	C	A	22
Chr4	26	T	C	12
Chr4	27	G	A	19
Chr4	47	A	C	22

5.5.3 Variation Annotation: SNP Effect Analysis (Filter and annotate identified variants)

The annotations include gene structure (exon, intron, utr etc), consequence of the variation (e.g. synonymous) and reported in dbSNP.

Usage: java -jar home/user/snpEff/snpEff.jar athalianaTair10 snps_tair10.txt

Chrom	Start	End	Zygosity	Variant Type	Reference	SNP Call	Total Reads	Variant Reads	Gene Symbol	Transcript	Consequence	AA Change
chr17	7579478	7579478	Hetero zygous	SNP	G	C	198	48	TP53	NM_000546	NON_SYNONYMOUS_CODING	A->G
chr17	7579478	7579478	Hetero zygous	SNP	G	C	198	48	TP53	NM_001126114	NON_SYNONYMOUS_CODING	A->G
chr17	7579478	7579478	Hetero zygous	SNP	G	C	198	48	TP53	NM_001126112	NON_SYNONYMOUS_CODING	A->G
chr17	7579478	7579478	Hetero zygous	SNP	G	C	198	48	TP53	NM_001126113	NON_SYNONYMOUS_CODING	A->G
chr4	178355579	178355579	Hetero zygous	SNP	C	G	331	120	AGA	NM_000027	NON_SYNONYMOUS_CODING	A->P
chr4	178355579	178355579	Hetero zygous	SNP	C	G	331	120	AGA	NM_001171988	NON_SYNONYMOUS_CODING	A->P
chr11	17418477	17418477	Homozygous	SNP	C	A	556	554	ABCC8	NM_000352	NON_SYNONYMOUS_CODING	A->S
chr20	25259006	25259006	Hetero zygous	SNP	G	T	90	51	PYGB	NM_002862	NON_SYNONYMOUS_CODING	A->S
chr3	38640439	38640439	Hetero zygous	SNP	C	A	471	217	SCN5A	NM_000335	NON_SYNONYMOUS_CODING	A->S
chr3	38640439	38640439	Hetero zygous	SNP	C	A	471	217	SCN5A	NM_001099405	NON_SYNONYMOUS_CODING	A->S
chr3	38640439	38640439	Hetero zygous	SNP	C	A	471	217	SCN5A	NM_001160161	NON_SYNONYMOUS_CODING	A->S
chr3	38640439	38640439	Hetero zygous	SNP	C	A	471	217	SCN5A	NM_001099404	NON_SYNONYMOUS_CODING	A->S
chr3	38640439	38640439	Hetero zygous	SNP	C	A	471	217	SCN5A	NM_001160160	NON_SYNONYMOUS_CODING	A->S
chr3	38640439	38640439	Hetero zygous	SNP	C	A	471	217	SCN5A	NM_198056	NON_SYNONYMOUS_CODING	A->S
chr1	216595306	216595306	Homozygous	SNP	C	T	790	789	USH2A	NM_007123	NON_SYNONYMOUS_CODING	A->T
chr1	216595306	216595306	Homozygous	SNP	C	T	790	789	USH2A	NM_206933	NON_SYNONYMOUS_CODING	A->T
chr10	73501556	73501556	Homozygous	SNP	G	A	627	626	CDH23	NM_022124	NON_SYNONYMOUS_CODING	A->T
chr17	10541515	10541515	Hetero zygous	SNP	C	T	46	25	MYH3	NM_002470	NON_SYNONYMOUS_CODING	A->T
chr3	53535667	53535667	Hetero zygous	SNP	G	A	12	2	CACNA1D	NM_001128839	NON_SYNONYMOUS_CODING	A->T
chr3	53535667	53535667	Hetero zygous	SNP	G	A	12	2	CACNA1D	NM_001128840	NON_SYNONYMOUS_CODING	A->T

5.6 Genomic & Exome Coverage

5.6.1 Coverage Convert one or more .sam alignment files to .wig

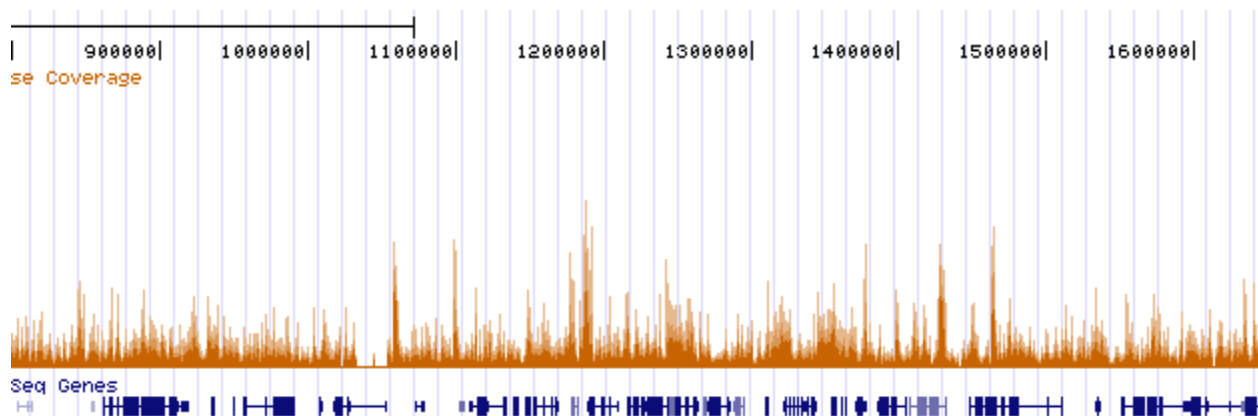
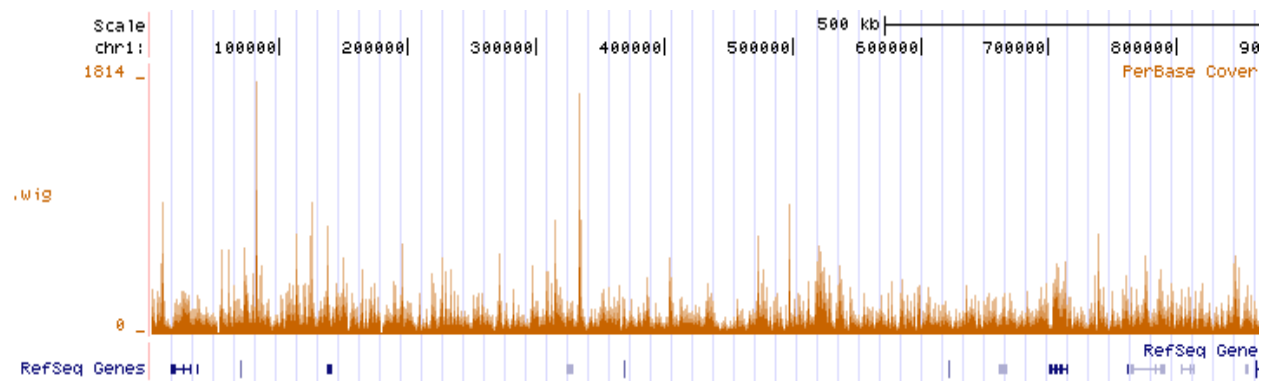
The output .wig file can be loaded and viewed as wiggle plot in various genome browsers, wiggle plots shows the coverage on per-base resolution in the genome

Usage: home/user/installDir/seqgene2_4/sam2wig.py input1.sam input2.sam
... output.wig convert one .sam file to .wig file or aggregate multiple .sam file to .wig file

output.wig snapshot:

```
track type=bedGraph name="ath.wig" description="PerBase Coverage"
```

```
chr1 1 2 0
chr1 2 7 2
chr1 7 8 3
chr1 8 19 4
chr1 19 20 8
chr1 20 23 12
chr1 23 26 14
chr1 26 27 15
chr1 27 30 16
chr1 30 32 17
chr1 32 35 18
chr1 35 39 20
chr1 39 41 21
chr1 41 44 22
chr1 44 47 27
chr1 47 49 29
chr1 49 50 30
chr1 50 51 31
```



5.7 Generate coverage quality report per exon

The report includes the coverage statistics for exon capture experiments. The percentage of reads that covers target exons (specificity) and coverage quantile across all exons (sensitivity). The report also calculate the percentage of exon regions that are not well covered (below the DEFECTIVE_CUT threshold).

Usage: home/user/installDir/seqgene2_4/exon_qc.py [options] input.wig output.qc

Options:

- h, --help show this help message and exit
- o ORGANISM, --organism=ORGANISM organism: human, mouse, arabidopsis, rat, hg18 etc.. see seqgene.config for example
[default: human]
- r RL, --readlength=RL read length [default: 60]
- e EXTEND_EXON, --extend_exon=EXTEND_EXON extend exon boundary by length
[default: 0]
- d DEFECTIVE_CUT, --defective_cut=DEFECTIVE_CUT (minmum coverage to be considered a defective segment [default: 10]

chr	strand	exon_start	exon_end	transcript	symbol	exon_number	deft_length_lt_5	Total length	per_defective	coverage
2	+	17207490	17207698	AT2G41260.1	M17	5	0	209	0	34308.196
2	+	17207490	17207698	AT2G41260.2	M17	4	0	209	0	34308.196
3	+	8011904	8011724	AT3G22640.1	PAP85	7	0	181	0	37900.536
4	+	13612328	13612563	AT4G27160.1	AT2S3	3	0	236	0	136048.496
5	+	13852611	13852390	AT5G35660.1	AT5G35660	3	0	222	0	109917.423
5	+	17756462	17756264	AT5G44120.3	CRU1	6	0	199	0	117741.643
5	+	17756462	17756264	AT5G44120.2	CRU1	6	0	199	0	117741.643
5	+	17756462	17756264	AT5G44120.1	CRU1	5	0	199	0	117741.643
5	+	22238642	22238358	AT5G54740.1	SESA5	3	0	285	0	87143.347
5	+	26518513	26518297	AT5G66400.1	RAB18	4	0	217	0	299209.691
5	+	26518513	26518277	AT5G66400.2	RAB18	4	0	237	0	274816.190

6. Transcriptome sequencing - a case study in *Piper*

Johnson K. George, IISR

Next-generation sequencing (NGS) technologies are increasingly applied in many organisms, including non model plants that are of economic importance. Illumina sequencing is one among the most used NGS technologies and have been shown to produce optimal results at reasonable costs. Application of Illumina sequencing technology to characterize the transcriptome of *Piper* species of economical relevance for which no genomic resource is available is described here. The basic goal of the study was to learn about *Piper*–*Phytophthora* interactome. In the area of plant-pathogen interactions, transcript profiling has been providing unparalleled perception into the mechanisms underlying gene-for-gene resistance and basal defense, host vs nonhost resistance, etc. among many others.

Transcriptome sequence assembly and analysis was done to facilitate a system-wide approach to study *Piper*-*Phytophthora* interactions with special emphasis on the identification of genes involved in resistance to the oomycete. Enhancing the genomic resources in *Piper* was another important objective of the study. The sequencing analysis was done with transcriptomes of *Piper colubrinum* and *Piper nigrum* (Black pepper Var. *IISR Shakthi*) leaves, challenge inoculated with *Phytophthora capsici*. In the present study, the transcriptome (mRNA) of *Piper* samples were sequenced with short reads on Illumina Genome Analyzer II platform. Only the RNA samples with 260 of 280 ratio from 1.9 to 2.1, 260 of 230 ratio from 2.0 to 2.5 and RIN (RNA integrity number) more than 8.0, were used for the analysis. The sequencing feature of the paired-end sequencing technology platform used here, yielded 2×75 bp independent reads from either end of a DNA fragment. The sequence data was filtered for low-quality reads at high stringency (reads with Phred quality score of equal to or less than 20) and reads containing primer/adaptor sequence. Assembly of Velvet followed by Oases yields better contigs/transcripts. The Oases program has been developed specifically for the de novo assembly of transcriptomes using short reads, which takes the assembly generated by Velvet (at different k-mer length) as input and exploits the read sequence and pairing information to produce transcript isoforms. The assembly resulted in contigs of at least 100 bp length. From the total number of 43121760 reads 32433 contigs were assembled (at k= 47) in case of *P. colubrinum* and for *Piper nigrum* sample, 70695884 reads were used at k= 53 to assemble into 90981 contigs. The assembly resulted in a total of 62619 and 101284 transcripts in case of *P. colubrinum* and *P. nigrum* samples (**Table 1**).

Item	Sample	
	<i>Piper Colubrinum</i>	<i>Piper nigrum</i>
Sequence File Size	37.70 MB	76.06 MB

Maximum Sequence Length	15769	10479
Minimum Sequence Length	100	100
Average Sequence Length	567.844	721.922
No. of Sequences	62619	101284
Total Sequences Length	35557875	73119148
Total Number of Non-ATGC Characters	1316	1090
Percentage of Non-ATGC Characters	0.00004	0.00001

Table 1: Summary of data generated for *Piper* transcriptome

6.1 Gene function annotation

BLAST searches against Plant (21 plant species including *Piper*), *Physcomitrella patens* (Moss) and *Phytophthora* gene databases were utilized for similarity searches and assigning gene function. BLAST hits to mRNA and protein databases of different plant species and *Phytophthora* were examined and the significant hits were identified at different E-value cut-offs. Out of 62619 transcripts from *Piper colubrinum* sample, transcripts, 22921 were annotated and 42835 out of 101284 transcripts were annotated in case of *Piper nigrum* sample.

Both *Piper colubrinum* and *Piper nigrum* transcripts showed maximum hit corresponding to mRNA and protein databases with *Vitis vinifera* (wine grape) sequences (56620), followed by *Populus trichocarpa* (Poplar) sequences (51332) indicating closer relationship of magnoliids (order to which *Piper* belong to) with eudicots. Magnoliids are considered one of the largest clades of early diverging angiosperms and it is hypothesized that the magnoliids are sister to a large clade that includes both monocots and eudicots.

Gene Ontology assignment programs for functional categorization of those annotated unigenes were done based on similarity with *Arabidopsis* sequences. Gene Ontology provides a structured and controlled vocabulary to describe gene products according to three ontologies: molecular function, biological process, and cellular component. Multiple assignments is also possible for a given protein within a single ontology and different isoforms were considered separately during analysis. In the case of *P. colubrinum* transcripts, 3160 were characterized under molecular functions (GOMF), 5866 under biological process (GOBP) and 2893 under cellular components (GOCC) category. Similarly, *P. nigrum* transcripts were able to map 3469 to molecular functions, 6549 to biological processes and 3419 to cellular component category. The genes involved in other important biological processes such as response to abiotic and biotic stimulus/stress, transport, transcription and signal transduction, were also identified through GO annotations. Broadly, the putative orthologs of genes involved in various pathways and cellular processes were found in both the transcriptomes. List of stress induced genes (**Table 2**), genes related to secondary metabolism, metabolic process, signal transduction etc.

aconitase	glutathione peroxidase
aldehyde dehydrogenase	glutathione reductase
Ascorbate peroxidase	Glutathione s-transferase
aspartyl protease family	glutathione synthetase
beta 1,3-glucanase	glyoxylate reductase (NADP)
beta-carotene hydroxylase (BCH)	isoflavone 7-O-methyltransferase
Catalase	L-galactose dehydrogenase (L-GalDH gene
cathepsin B-like cysteine protease	lipoxigenase (DOX1)
CBL-interacting protein kinase 10 (CIPK10)	NADH-plastoquinone oxidoreductase
Chitinases	papain-like cysteine proteinase-like protein
cinnamoyl CoA reductase	p-coumaroyl shikimate 3'- hydroxylase
cinnamyl alcohol dehydrogenase	peroxidase
Cu/Zn superoxide dismutase	peroxisomal betaine-aldehyde dehydrogenase
cysteine proteinase	polygalacturonase inhibiting protein- pgip1 gene
dehydratase	proteasome alpha subunit
dehydration-induced proteins (dehydrins)	pyrroline-5-carboxylate reductase
delta 1-pyrroline-5-carboxylate synthetase	pyrroline-5-carboxylate synthetase
Early response to dehydration (ERD2)	serine palmitoyltransferase
flavin-containing monooxygenase	thaumatin-like protein
galactinol synthase	

Table 2: Genes related to response to stress identified with the transcriptome sequences of *Piper colubrinum* and *Piper nigrum* leaves challenged with *Phytophthora*

The identified stress induced genes include catalase, chitinase class I and VII, glutathione-S-transferase, peroxidase, beta 1,3-glucanase, Cu/Zn superoxide dismutase, manganese superoxide dismutase, MAP kinase, osmotin etc. Among the genes, those identical to genes involved in secondary metabolism were, chalcone isomerase, Chalcone synthase, cinnamate 4-hydroxylase, cinnamoyl-CoA reductase, geranyl geranyl pyrophosphate synthase, hmg-CoA reductase, lycopene beta cyclase, phenylalanine ammonia lyase, p-coumaroyl shikimate 3'- hydroxylase and Transaldolase. A variety of transcription factors and genes involved in primary metabolism with significant similarity to those characterized in other plants were also identified in both transcriptomes.

6.2 Expression of *Phytophthora* genes

Expression of *Phytophthora* genes in *Planta* were also examined and when both transcriptomes were considered together maximum number of genes from *Phytophthora infestans*, followed by *P. capsici*. This could be because maximum annotated gene list available in case of *P. infestans*, compared to that of *P. capsici*. Limited number of genes from *P. palmivora* and *P. tropicalis*

were also annotated. Some of the matching sequences from different *Phytophthora* species were, for catalase, alfa and beta-tubulins, heat shock proteins, enolase, endo-1, 4-beta glucanase, ubiquitin family proteins, Rab1 family GTPase, calmodulin and members of the Ras super family of monomeric GTP-binding proteins, essential in specific steps of vesicle transport and secretion.

6.3 Gene specific transcript analysis

Transcripts with similarity to selected reference genes were analyzed from the transcriptome data. Length, Identity (%), Alignment length and E- Value of the contigs with significant hits to selected reference genes from plants are given in **table 3**. The selected sequences include those gene already identified in *Piper* and those genes involved in stress resistance. The length of the contigs corresponding to specific genes varied in different samples. Contigs similar to partially sequenced R gene related sequences and Malate dehydrogenase were found only in *P. colubrinum* sample. This could be because of high variation in similar genes of *P. nigrum*.

Reference gene	<i>Piper colubrinum</i>				<i>Piper nigrum</i>			
	Contig length	Identity (%)	Alignment length	E- Value	Contig length	Identity (%)	Alignment length	E- Value
Aquaporin	2094	99.63	267	2e-149	1229	82.65	98	1e-10
Osmotin	297	99.35	155	4e-81	318	94.53	201	4e-87
betaine-aldehyde dehydrogenase	1698	77.27	726	3e-27	1726	77.61	603	2e-25
Cu/Zn superoxide dismutase	865	88.29	401	2e-72	797	82.93	375	5e-64
Mitogen-activated protein kinase (MAPK)	1798	77.89	995	3e-61	2854	78.35	485	3e-31
R gene related sequence (previously identified from <i>P. colubrinum</i>)	3008	98.76	242	1e-129	-	-	-	
bZIP transcription factor	2053	76.62	633	7e-16	1797	78.65	342	3e-21
beta-1,3-glucanase-like gene	1074	97.96	490	0	628	93.21	265	3e-108

Calmodulin	1742	83.79	1497	3e-61	1801	82.93	375	1e-63
Catalase	1576	78.83	1162	4e-94	1353	79.31	1020	8e-98
Geranylgeranyl transferase	1124	79.9	398	3e-35	542	81.46	329	1e-39
Heat shock protein-70	2401	79.48	1433	7e-146	1857	78.33	1269	1e-88
Malate dehydrogenase	2304	77.95	1111	1e-70	-	-	-	
WRKY	1620	97.92	96	8e-43	977	100	40	9e-17
Alpha amylase	2206	98.24	227	6e-109	2721	94.95	198	5e-82

Table 3: Gene specific contigs/ transcripts identified from *P. colubrinum* and *P. nigrum* transcriptome

6.4 Comparative gene expression analysis in *Piper colubrinum* and *P. nigrum*

Quantification of differential gene transcription data was done based on average read depth of specific gene sequences from transcriptome data of *Piper colubrinum* and *P. nigrum* challenged with *Phytophthora capsici*. The gene sequences of significant similarity to some genes already identified in *Piper* and other plants were used in the study. Significant differences in gene isoform expression levels was observed in the samples examined (**table 4**). Extensive transcriptional activity (based on read depth) of defense related genes viz., Osmotin and beta-1,3-glucanase revealed was observed in *Piper colubrinum* leaf sample, challenged with *Phytophthora capsici*. The average read depth of different genes ranged from 0 to 2016108 in both the samples. R gene related sequence of *P. colubrinum* had a read depth of 6161.33.

Gene of interest	Expression in <i>Piper colubrinum</i> *	Expression in <i>Piper nigrum</i> *
ACC oxidase	7.57	0.00
PISTILLATA-like protein PI	13.33	0.00
APETALA3-like protein AP3-2	41.38	0.00
heat shock protein-70 cognate protein (ERD2)	104.85	3.88
Cinnamoyl CoA reductase	531.15	37020.49
Alpha amylase	2558.75	5379.58
WRKY transcription factor	5197.90	14193.71
R gene related sequence (<i>P. colubrinum</i>)	6161.33	0.00
Hydroxyproline-rich glycoprotein	11175.07	26616.99
Peroxidase	30732.80	183601.93

beta-1,3-glucanase	66399.50	2825.04
Aquaporin	75066.67	6042.32
Osmotin	2016108.51	394.81

*Based on average depth

Table 4: Expression data for specific genes in the *Piper colubrinum* and *P. nigrum* transcriptome

6.5 Identification of Resistance Gene Analogues (RGAs) in the transcriptomes

In most cases, R genes are of the NBS-LRR class of R genes, encoding receptor-like proteins that most likely recognize an avirulence factor and trigger a defense response. Rgene analogs (RGAs) share several common motifs that are highly conserved. These include the P loop (phosphate-binding domain), the kinase-2 motif, and the GLPL motif. These motifs have been widely utilized for the identification or the cloning of resistance genes.

Nine NBS related transcripts from *Piper colubrinum* was found and was related to sequences of *Arabidopsis thaliana*, *Populus trichocarpa*, *Brassica napus*, *Glycine max* and *Hordeum vulgare* (**Table 5**). Similarly, about 15 transcripts from *Piper nigrum* was found to be related to NBS type of resistance genes.

S. No	Transcript No. (PN)	Length (bp)	Most similar R gene related sequence	coverage	E-value
1	Locus_22875	143	<i>Vitis vinifera</i> probable disease resistance protein At5g63020-like mRNA	94%	3e-10
2	Locus_51895	127	<i>Brassica rapa subsp. pekinensis</i> isolate BrCNL5 disease resistance protein gene, complete cds	99%	4e-10
3	Locus_14276	178	<i>Vitis vinifera</i> putative disease resistance RPP13-like protein 1 mRNA	79%	2e-07
4	Locus_19163	134	<i>Vitis vinifera</i> putative disease resistance protein mRNA	99%	2e-08
5	Locus_34918	148	<i>Vitis vinifera</i> putative disease resistance protein mRNA	99%	2e-08
6	Locus_52055	183	<i>Ipomoea batata</i> isolate S1_C09 resistance gene analog genomic sequence	83%	6e-09
7	Locus_11301	1086	<i>Populus trichocarpa</i> NBS resistance protein, mRNA	38%	3e-51
8	Locus_48409	127	<i>Solanum demissum</i> isolate 286O21c disease resistance protein R3a-like protein	97%	1e-10

			pseudogene, complete sequence		
9	Locus_53603	122	<i>Medicago truncatula</i> TIR-NBS-LRR type disease resistance protein mRNA, complete cds	91%	6e-12

Table 5: Data on the NBS-LRR type of resistance gene related transcripts (putative) identified from *P. colubrinum*

6.6 Transcript analysis of *Piper colubrinum* for WRKY related sequences

Transcription factors (TFs) represent key proteins that bind to specific DNA sequences and regulate gene expression. TFs are represented by various multigene families and are highly conserved in eukaryotic organisms, especially plants.

The WRKY proteins are a super family of transcription factors involved in the regulation of various physiological programs including pathogen defense. It is believed that these proteins are regulatory transcription factors with a binding preference for the W box, but with the potential to differentially regulate the expression of a variety of target genes. WRKY proteins was found to bind to the W box, which is also found in the promoters of Many plant defense genes.

Two related transcripts based on the match with previously identified 143 bp fragment of the WRKY gene was found in the transcriptome. The length of these transcripts were 1528 and 1620 bp each . Comparative analysis of the sequences using Clustal W reveled that the one of these sequences has sequence is having an intron of 92 nucleotides long.

Full length of the gene was deduced and found to have a coding sequence of 303 amino acids corresponding to 909 nucleotides. NCBI protein blast of this sequence showed maximum similarity with WRKY gene from *Dimocarpus longan* with an E value of 3e-90. Strong match with WRKY sequences from *Populus trichocarpa*, *Vitis vinifera*, *Solanum lycopersicum* etc were also found. Domain hits for WRKY genes (E value: 6.33e -33) were found when conserved domain search through NCBI was conducted. InterPro Scan (EBI) also resulted in the identification of WRKY – DNA binding domains (sequence specific DNA binding transcription factor activity) and showed strong match. Conserved Domain Search (NCBI) also reveled strong similarity with WRKY proteins from *Populus trichocarpa*, *Vitis vinifera*, *Arabidopsis thaliana*, *Sorghum bicolor*, *Oryza sativa* etc.

A large number of SNPs were also identified in *Piper colubrinum* (50972 nos.) and in *Piper nigrum* (231680 nos.) transcriptome. The data need to be further checked and to be confirmed utilizing wet laboratory experiments. High occurrence of SNPs in *Piper nigrum* could be because

of polyploid (tetraploid) nature of black pepper. These SNPs, once confirmed, is expected to provide a valuable resource for future studies on genetic linkage mapping and the analysis of interesting traits in black pepper. Our results demonstrate the utility of NGS technologies as starting point for the development of genomic tools in non model but one of the most important spice crops.

7. Comparative Genomics

7.1 What is comparative genomics?

Comparative genomics is an exciting new field of biological research in which the genome sequences of different species - human, mouse and a wide variety of other organisms from yeast to chimpanzees - are compared. Comparative genomics is the study of the relationship of genome structure and function across different biological species or strains. Comparative genomics is an attempt to take advantage of the information provided by the signatures of selection to understand the function and evolutionary processes that act on genomes. While it is still a young field, it holds great promise to yield insights into many aspects of the evolution of modern species. The sheer amount of information contained in modern genomes (3.2 gigabases in the case of humans) necessitates that the methods of comparative genomics are automated. Gene finding is an important application of comparative genomics, as is discovery of new, non-coding functional elements of the genome.

By comparing the finished reference sequence of the human genome with genomes of other organisms, researchers can identify regions of similarity and difference. This information can help scientists better understand the structure and function of human genes and thereby develop new strategies to combat human disease. Comparative genomics also provides a powerful tool for studying evolutionary changes among organisms, helping to identify genes that are conserved among species, as well as genes that give each organism its unique characteristics.

Human FOXP2 gene and evolutionary conservation is shown in and multiple alignment (at bottom of figure) in this image from the UCSC Genome Browser. Note that conservation tends to cluster around coding regions (exons).

Comparative genomics exploits both similarities and differences in the proteins, RNA, and regulatory regions of different organisms to infer how selection has acted upon these elements. Those elements that are responsible for similarities between different species should be conserved through time (stabilizing selection), while those elements responsible for differences among species should be divergent (positive selection). Finally, those elements that are unimportant to the evolutionary success of the organism will be unconserved (selection is neutral).

One of the important goals of the field is the identification of the mechanisms of eukaryotic genome evolution. It is however often complicated by the multiplicity of events that have taken place throughout the history of individual lineages, leaving only distorted and superimposed traces in the genome of each living organism. For this reason comparative genomics studies of small model organisms (for example the model *Caenorhabditiselegans* and closely related *Caenorhabditisbriggsae*) are of great importance to advance our understanding of general mechanisms of evolution.

Having come a long way from its initial use of finding functional proteins, comparative genomics is now concentrating on finding regulatory regions and siRNA molecules. Recently, it has been discovered that distantly related species often share long conserved stretches of DNA that do not appear to code for any protein (see conserved non-coding sequence). One such ultra-conserved region, that was stable from chicken to chimp has undergone a sudden burst of change in the human lineage, and is found to be active in the developing brain of the human embryo.

Computational approaches to genome comparison have recently become a common research topic in computer science. A public collection of case studies and demonstrations is growing, ranging from whole genome comparisons to gene expression analysis. This has increased the introduction of different ideas, including concepts from systems and control, information theory, strings analysis and data mining. It is anticipated that computational approaches will become and remain a standard topic for research and teaching, while multiple courses will begin training students to be fluent in both topics.

7.2 Benefits of comparative genomics

Dramatic results have emerged from the rapidly developing field of comparative genomics. Comparison of the fruit fly genome with the human genome reveals that about sixty percent of genes are conserved (Adams *et al.* 2000). That is, the two organisms appear to share a core set of genes. Researchers have also found that two-thirds of human genes known to be involved in cancer have counterparts in the fruit fly.

Using computer-based analysis to zero in on the genomic features that have been preserved in multiple organisms over millions of years, researchers will be able to pinpoint the signals that control gene function, which in turn should translate into innovative approaches for treating human disease and improving human health. In addition to its implications for human health, comparative genomics may benefit the broader animal world and ecological studies as well. As sequencing technology grows easier and less expensive, it will find wide applications in agriculture, biotechnology, and zoology as a tool to tease apart the often-subtle differences among animal and plant species. Such efforts might also lead to the rearrangement of our understanding of some branches of the evolutionary "tree of life," as well as point to new strategies for conserving rare and endangered species.

7.3 Why is there an increased interest in genomics?

Although living creatures look and behave in many different ways, all of their genomes consist of DNA, the chemical chain that makes up the genes that code for thousands of different kinds of proteins. Precisely which protein is produced by a given gene is determined by the sequence in which four chemical building blocks - adenine (A), thymine (T), cytosine (C) and guanine (G) - are laid out along DNA's double-helix structure.

In order for researchers to use an organism's genome most efficiently in comparative studies, data about its DNA must be in large, contiguous segments, anchored to chromosomes and,

ideally, fully sequenced. Furthermore, the data needs to be organized to allow easy access for researchers using sophisticated computer software to conduct high-speed analyses.

The successful completion of the Human Genome Project in April 2003 has demonstrated that large-scale sequencing projects can generate high-quality data at a reasonable cost. As a result, the interest in sequencing the genomes of many other organisms has risen dramatically.

7.4 What other genomes have been sequenced?

In addition to sequencing the 3 billion letters in the human genetic instruction book, researchers involved in the Human Genome Project have already sequenced the genomes of a number of important model organisms that are commonly used as surrogates in studying human biology. These are the chimpanzee, the mouse, the rat, two puffer fish, two fruit flies, two sea squirts, two roundworms, baker's yeast and the bacterium *Escherichia coli*. Currently, sequencing centers supported by the National Human Genome Research Institute (NHGRI) of the National Institutes of Health (NIH) are close to completing working drafts of the chicken, the dog, the honey bee, the sea urchin and a set of four fungi. In the summer of 2003, the centers also began sequencing the genome of the rhesus macaque monkey, and many other organisms are in the sequencing pipeline.

7.5 Major outcomes of comparative genomics

The rapidly emerging field of comparative genomics has already yielded dramatic results. For example, a March 2000 study comparing the fruit fly genome with the human genome discovered that about 60 percent of genes are conserved between fly and human. Or, to put it simply, the two organisms appear to share a core set of genes.

Researchers have found that two-thirds of human genes known to be involved in cancer have counterparts in the fruit fly. Even more surprisingly, when scientists inserted a human gene associated with early-onset Parkinson's disease into fruit flies, they displayed symptoms similar to those seen in humans with the disorder, raising the possibility the tiny insects could serve as a new model for testing therapies aimed at Parkinson's.

More recently, a comparative genomic analysis of six species of yeast prompted scientists to significantly revise their initial catalog of yeast genes and to predict a new set of functional elements thought to play a role in regulating genome activity.

7.6 How Are Genomes Compared?

A simple comparison of the general features of genomes such as genome size, number of genes, and chromosome number presents an entry point into comparative genomic analysis. Data for several fully-sequenced model organisms is shown in Table 1. The comparisons highlight some striking findings. For example, while the tiny flowering plant *Arabidopsis thaliana* has a smaller genome than that of the fruit fly *Drosophila melanogaster* (157 million base pairs v. 165 million base pairs, respectively) it possesses nearly twice as many genes (25,000 v. 13,000). In fact A.

thaliana has approximately the same number of genes as humans (~25,000). Thus, a very early lesson learned in the "genomic era" is that genome size does not correlate with evolutionary status, nor is the number of genes proportionate to genome size.

Table 1. Comparative genome sizes of humans and other model organisms

Organism	Estimated size (base pairs)	Chromosome number	Estimated gene number
Human (<i>Homo sapiens</i>)	3 billion	46	~25,000
Mouse (<i>Mus musculus</i>)	2.9 billion	40	~25,000
Fruit fly (<i>Drosophila melanogaster</i>)	165 million	8	13,000
Plant (<i>Arabidopsis thaliana</i>)	157 million	10	25,000
Roundworm (<i>Caenorhabditis elegans</i>)	97 million	12	19,000
Yeast (<i>Saccharomyces cerevisiae</i>)	12 million	32	6,000
Bacteria (<i>Escherichia coli</i>)	4.6 million	1	3,200

Table 1 : Comparative genome sizes of humans and other model organisms

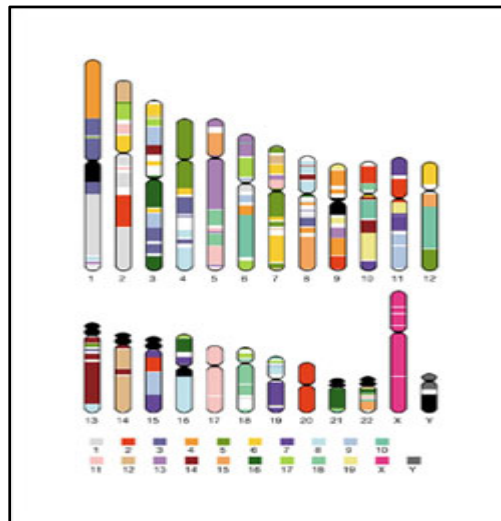


Figure 1: Conserved segments in the human and mouse genome

Human chromosomes, with segments containing at least two genes whose order is conserved in the mouse genome as color blocks. Each color corresponds to a particular mouse chromosome. Centromeres, are arranged in similar blocks in different species. The nature and extent of conservation of synteny differs substantially among chromosomes. For example, the X chromosomes are represented as single, reciprocal syntenic blocks. Human chromosome 20 corresponds entirely to a portion of mouse chromosome 2, with nearly perfect conservation of order along almost the entire length, disrupted only by a small central segment. Human

chromosome 17 corresponds entirely to a portion of mouse chromosome 11. Other chromosomes, however, show evidence of more extensive inter chromosomal rearrangement. Results such as these provide an extraordinary glimpse into the chromosomal changes that have shaped the mouse and human genomes since their divergence from a common ancestor 75–80 million years ago.

Comparison of discrete segments of genomes is also possible by aligning homologous DNA from different species. An example of such an alignment is shown in Figure 2, where a human gene (pyruvate kinase: PKLR) and the corresponding PKLR homologs from macaque, dog, mouse, chicken, and zebrafish are aligned. Regions of high DNA sequence similarity with human across a 12-kilobase region of the PKLR gene are plotted for each organism. Notice the high degree of sequence similarity between human and macaque (two primates) in both PKLR exons (blue) as well as introns (red) and untranslated regions (light blue) of the gene. In contrast, the chicken and zebra fish alignments with human only show similarity to sequences in the coding exons; the rest of the sequence has diverged to a point where it can no longer be reliably aligned with the human DNA sequence. Using such computer-based analysis to zero in on the genomic features that have been preserved in multiple organisms over millions of years, researchers are able to locate the signals that represent the location of genes, as well as sequences that may regulate gene expression. Indeed, much of the functional parts of the human genome have been discovered or verified by this type of sequence comparison (Lander *et al.* 2001) and it is now a standard component of the analysis of every new genome sequence.

subcentromeric heterochromatin of chromosomes 1, 9 and 16, and the repetitive short arms of 13, 14, 15, 21 and 22 are in black. (International Human Genome Sequencing Consortium; Lander, E. S. et al. 2001)

Finer-resolution comparisons are possible by direct DNA sequence comparisons between species. Figure 1 depicts a chromosome-level comparison of the human and mouse genomes that shows the level of synteny between these two mammals. Synteny is a situation in which genes

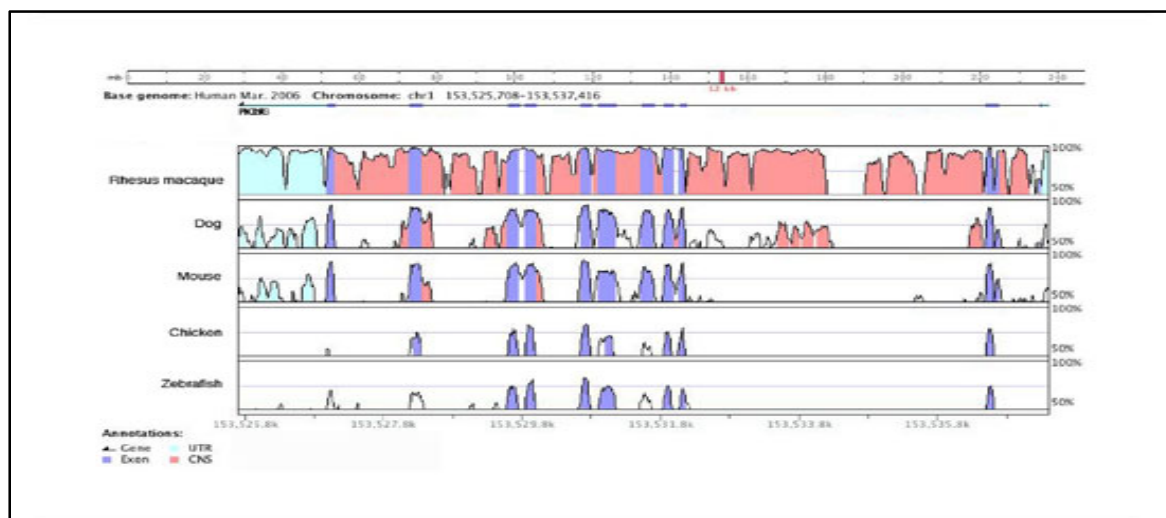


Figure 2: Human PKLR gene region compared to the macaque, dog, mouse, chicken, and zebra fish genomes

Numbers on the vertical axis represent the proportion of identical nucleotides in a 100-bp window for a point on the plot. Numbers on the horizontal axis indicate the nucleotide position from the beginning of the 12-kilobase human genomic sequence. Peaks shaded in blue correspond to the PKLR coding regions. Peaks shaded in light blue correspond to PKLR mRNA untranslated regions. Peaks shaded in red correspond to conserved non-coding regions (CNSs), defined as areas where the average identity is > 75%. Alignment was generated using the sequence comparison tool VISTA (<http://pipeline.lbl.gov>).

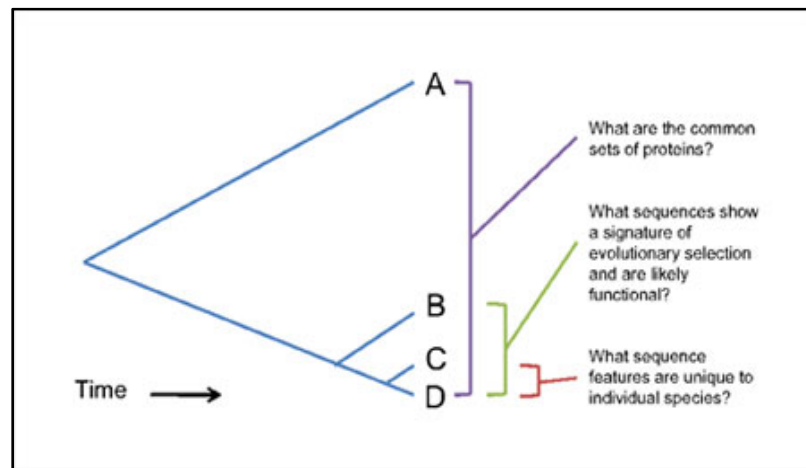


Figure 3: Comparisons of genomes at different phylogenetic distances serve to address specific questions.

We have learned from homologous sequence alignment that the information that can be gained by comparing two genomes together is largely dependent upon the phylogenetic distance between them. Phylogenetic distance is a measure of the degree of separation between two organisms or their genomes on an evolutionary scale, usually expressed as the number of accumulated sequence changes, number of years, or number of generations. The distances are often placed on phylogenetic trees, which show the deduced relationships among the organisms (Figure 3). The more distantly related two organisms are, the less sequence similarity or shared genomic features will be detected between them. Thus, only general insights about classes of shared genes can be gathered by genomic comparisons at very long phylogenetic distances (e.g., over one billion years since their separation). Over such very large distances, the order of genes and the signatures of sequences that regulate their transcription are rarely conserved.

At closer phylogenetic distances (50–200 million years of divergence), both functional and non-functional DNA is found within the conserved segments. In these cases, the functional sequences will show signatures of selection by virtue of their sequences having changed less, or more slowly than, non-functional DNA. Moreover, beyond the ability to discriminate functional from non-functional DNA, comparative genomics is also contributing to the identification of general

classes of important DNA elements, such as coding exons of genes, non-coding RNAs, and some gene regulatory sites. In contrast, very similar genomes separated by about 5 million years of evolution (such as human and chimpanzee) are particularly useful for finding the sequence differences that may account for subtle differences in biological form. These are sequence changes under directional selection, a process whereby natural selection favors a single phenotype and continuously shifts the allele frequency in one direction. Comparative genomics is thus a powerful and promising approach to biological discovery that becomes more and more informative as genomic sequence data accumulate.

7.7 Practical- VISTA Tools for comparative genomics

Comparison of DNA sequences from different species is a fundamental method for identifying functional elements in genomes. VISTA is a comprehensive suite of programs and databases for comparative analysis of genomic sequences. There are two ways of using VISTA - you can submit your own sequences and alignments for analysis (VISTA servers) or examine pre-computed whole-genome alignments of different species.

VISTA tools are created to assist biologists in carrying out this task. First VISTA server at <http://www-gsd.lbl.gov/vista/> was launched in the summer of 2000 and was designed to align long genomic sequences and visualize these alignments with associated functional annotations. Currently the VISTA site includes multiple comparative genomics tools and provides users with rich capabilities to browse pre-computed whole-genome alignments of large vertebrate genomes and other groups of organisms with VISTA Browser, to submit their own sequences of interest to several VISTA servers for various types of comparative analysis and to obtain detailed comparative analysis results for a set of cardiovascular genes. We illustrate capabilities of the VISTA site by the analysis of a 180 kb interval on human chromosome 5 that encodes for the kinesin family member 3A (*KIF3A*) protein.

7.7.1 Tools for comparative genomics

(i) mVISTA

Align and compare your sequences from multiple species mVISTA. mVISTA is designed to perform pairwise alignments of DNA sequences up to mega bases long from two or more species and to visualize these alignments together with annotations. AVID is the alignment engine behind mVISTA, and it allows the global alignment of DNA sequences of arbitrary length. To use mVISTA for comparative sequence analysis, two or more sequences in FASTA format (plain text only) or GenBank accession numbers together with a gene annotation file are submitted to the Web server.

(ii) rVISTA

rVISTA (regulatory VISTA) combines searching the major transcription factor binding site database TRANSFAC™ Professional from Biobase with a comparative sequence analysis. It can be used directly or through links in mVISTA, GenomeVISTA and VISTA Browser. The visualization program for rVISTA allows the user to look at binding sites for a single transcription factor and/or various combinations of transcription factor binding sites, which allows the user easily to examine the clustering of binding sites for factors that are believed to interact with one another. Both global (AVID) and local (BLASTZ) alignment algorithms are incorporated into rVISTA.

(iii) Genome VISTA

Compare your sequences with several whole genome assemblies. It will automatically find the ortholog, obtain the alignment and VISTA plot.

❖ wgVISTA

Align sequences up to 10Mb long (finished or draft) including microbial whole-genome assemblies.

❖ Phylo-Vista

The Phylo-VISTA program with its associated web server presents a novel method for the visualization and analysis of conservation in multiple sequence alignments by providing several significant extensions to VISTA tools. Analyze multiple DNA sequence alignments of sequences from different species while considering their phylogenetic relationships.

7.7.2 Installation and Usage

VISTA is an application that allows the visualization of long sequence alignments with annotation information. The VISTA program uses the file or files (to visualize several related alignments) produced by any procedure of global or local alignment (such as BLAST, Gap (GCG), etc.) of two DNA sequences and parsed by the user according to 'Alignment_file' format. In a short time we are planning to release the stand-alone global alignment software to use as the first step before visualization.

The VISTA plot is based on moving a user-specified window over the entire alignment and calculating the percent identity over the window at each base pair. The X-axis represents the base sequence; the Y-axis represents the percent identity. If the user supplies an annotation file, genes and exons are marked above the plot. The direction of genes is indicated by an arrow, while the

coding exons and UTRs are marked with rectangles of different color. Conserved regions are highlighted under the curve, with red indicating a conserved non-coding region and blue indicating a conserved exon. Conserved UTRs are colored turquoise. The colors can be modified by the user. A conserved region is defined with percentage and length cutoffs. Conserved segments with percent identity X and length Y are defined to be regions in which every contiguous sub-segment of length Y was at least X% identical to its paired sequence. These segments are merged to define the conserved regions.

VISTA can be configured for visualizing alignments of various lengths by changing several parameters: the number of pages on which the output appears, the number of frames per page, the window size, and the resolution at which the alignment is plotted. VISTA allows one to easily create figures for various documents. For simplicity it is also possible to specify only a subset of these parameters, with the rest being automatically calculated. VISTA also supports simultaneous visualization of several related alignments.

(i) Installation:

Create a directory and copy Vista.jar and retePDF2.jar to it. Then change your CLASSPATH environment variable to include references to these two files.

(ii) Example I (Windows):

1. mkdir c:\vista
2. copy source_path\Vista.jar c:\vista
3. copy source_path\retePDF2.jar c:\vista
4. set CLASSPATH=c:\vista\Vista.jar;c:\vista\retePDF2.jar

(iii) Example II (UNIX, csh/tcsh):

1. mkdir target_path/vista
2. cp source_path/Vista.jar target_path/vista
3. cp source_path/retePDF2.jar target_path/vista
4. setenv CLASSPATH "target_path/vista/Vista.jar:target_path/vista/retePDF2.jar"

(iv) Usage: java Vista [-options] plot_file

where "plot_file" is the name of a file containing plot parameters (for file format see Appendix III),

and options include:

- q turn on quiet mode
- d turn on debug mode

7.8 Further Reading

1. Stein LD, *et al.* (2003). "The genome sequence of *Caenorhabditis briggsae*: a platform for comparative genomics". *PLoS Biology* **1** (2): E45. doi:10.1371/journal.pbio.0000045. PMC 261899.PMID 14624247.
2. "Newly Sequenced Worm a Boon for Worm Biologists". *PLoS Biology* **1** (2): e4–e4. 2003. doi:10.1371/journal.pbio.0000044.edit
3. Bejerano, Gill; Michael Pheasant, Igor Makunin, Stuart Stephen, W James Kent, John S Mattick, David Haussler (2004-05-28). "Ultraconserved elements in the human genome". *Science (New York, N.Y.)* **304** (5675): 1321-1325. doi:10.1126/science.1098119. ISSN 1095-9203.Retrieved 2012-01-13.
4. Pollard, Katherine S.; Sofie R. Salama, Nelle Lambert, Marie-Alexandra Lambot, Sandra Coppens, Jakob S. Pedersen, Sol Katzman, Bryan King, Courtney Onodera, Adam Siepel, Andrew D. Kern, Colette Dehay, Haller Igel, Manuel Ares, Pierre Vanderhaeghen, David Haussler (2006). "An RNA gene expressed during cortical development evolved rapidly in humans". *Nature* **443** (7108): 167-172. doi:10.1038/nature05113. ISSN 0028-0836. Retrieved 2012-01-13.
5. Cristianini N and Hahn M (2006). *Introduction to Computational Genomics*.Cambridge University Press.ISBN 0-5216-7191-4.
6. Via, Allegra; Javier De Las Rivas, Teresa K. Attwood, David Landsman, Michelle D. Brazas, Jack A. M. Leunissen, Anna Tramontano, Maria Victoria Schneider (2011-10-27). "Ten Simple Rules for Developing a Short Bioinformatics Training Course". *PLoS Comput Biol* **7** (10): e1002245. doi:10.1371/journal.pcbi.1002245. Retrieved 2011-12-03.
7. Adams, M. D., Celniker, S. E. *et al.* The genome sequence of *Drosophila melanogaster*. *Science* **287**, 2185-2195 (2000).
8. Aparicio, S., Chapman, J. *et al.* Whole-genome shotgun assembly and analysis of the genome of *Fugu rubripes*. *Science* **297**, 1301-1310 (2002).
9. Blattner, F. R., Plunkett, G. *et al.* The complete genome sequence of *Escherichia coli* K-12. *Science* **277**, 1453-1462 (1997).
10. Blattner, F. R., Plunkett, G. *et al.* Sequence and comparative analysis of the chicken genome provide unique perspectives on vertebrate evolution. *Nature* **432**, 695-716 (2004).
11. Deha, P., Satou, Y. *et al.* The draft genome of *Ciona intestinalis*: insights into chordate and vertebrate origins. *Science* **298**, 2157-2167 (2002).
12. Elsik, C. G., Tellam, R. L. *et al.* The genome sequence of taurine cattle: a window to ruminant biology and evolution. *Science* **324**, 522-528 (2009).
13. Gibbs, R. A., Rogers, J. *et al.* Evolutionary and biomedical insights from the rhesus macaque genome. *Science* **316**, 222-234 (2007).
14. Gibbs, R. A., Weinstock, G. M. *et al.* Genome sequence of the Brown Norway rat yields insights into mammalian evolution. *Nature* **428**, 493-521 (2004).
15. Goffeau, A., Barrell, B. G. *et al.* Life with 6000 genes. *Science* **274**, 546, 563-547 (1996).
16. Jaillon, O., Aury, J. M. *et al.* Genome duplication in the teleost fish *Tetraodon lineatus* reveals the early vertebrate proto-karyotype. *Nature* **431**, 946-957 (2004).

17. Lander, E. S., Linton, L. M. *et al.* Initial sequencing and analysis of the human genome. *Nature* 409, 860-921 (2001).
18. Lander, E. S., Linton, L. M. *et al.* Initial sequence of the chimpanzee genome and comparison with the human genome. *Nature* 437, 69-87 (2005).
19. Lindblad-Toh, K., Wade, C. M. *et al.* Genome sequence, comparative analysis and haplotype structure of the domestic dog. *Nature* 438, 803-819 (2005).
20. Lindblad-Toh, K., Wade, C. M. *et al.* Insights into social insects from the genome of the honeybee *Apis mellifera*. *Nature* 443, 931-949 (2006).
21. Small, K. S., Brudno, M. *et al.* A haplome alignment and reference sequence of the highly polymorphic *Ciona savignyi* genome. *Genome Biol* 8, R41 (2007).
22. Sodergren, E., Weinstock, G. M. *et al.* The genome of the sea urchin *Strongylocentrotus purpuratus*. *Science* 314, 941-952 (2006).
23. Stein, L. D., Bao, Z. *et al.* The genome sequence of *Caenorhabditis briggsae*: a platform for comparative genomics. *PLoS Biol* 1, E45 (2003).
24. Stein, L. D., Bao Z. *et al.* Genome sequence of the nematode *C. elegans*: a platform for investigating biology. The *C. elegans* Sequencing Consortium. *Science* 282, 2012-2018 (1998).
25. Waterston, R.H., Lindblad-Toh, K., *et al.* Initial sequencing and comparative analysis of the mouse genome. *Nature* 420:520-562 (2002).
26. Watson, J.D., Crick, F.H. Molecular structure of nucleic acids; a structure for deoxyribose nucleic acid. *Nature* 171(4356):737-738 (1953).

APPENDIX

The targeted sequence-enrichment technologies for NGS

Technology	Approach	Platform	Website
Agilent/SureSelect	Array- and solution-based	Illumina/Roche/ABI	http://www.chem.agilent.com/
RainDance	Microdroplet-based	Illumina/Roche/ABI	http://www.raindancetechnologies.com/
NimbleGen/SeqCap/EZ Exome	Array- and solution-based	Illumina/Roche/ABI	http://www.nimblegen.com/products/seqcap/index.html
Febit/HybSelect	Microarray-based	Illumina/Roche/ABI	http://www.febit.com/microarray-sequencing/index.cfm
Fluidigm	PCR-based	Illumina/Roche/ABI	http://www.fluidigm.com/targeted-resequencing.html
Mycroarray/Myselect	Solution-based	Illumina/Roche/ABI	http://www.mycroarray.com/products/myselect.html
LC Sciences	Microarray-based	Illumina/Roche/ABI	http://www.lcsciences.com/applications/genomics/
Qiagen/SeqTarget	Long-range PCR-based	Illumina/Roche/ABI	http://www.qiagen.com/products/seqtargetsystem.aspx
Illumina/TruSeq	Solution-based	Illumina/Roche/ABI	http://www.illumina.com/applications.ilmn

The alignment, assembly and utility bioinformatics tools for NGS.

Program	Function	Platform	Website
De novo assembly			
Abyss	Alignment/assembly	Illumina	http://www.bcgsc.ca/platform/bioinfo/software/abyss
ALLPATHS	Alignment/assembly	Illumina	http://www.broadinstitute.org/science/programs/genome-biology/crd
AMOScmp	Alignment/assembly	Roche	http://sourceforge.net/projects/amos/files/
ARACHNE	Alignment/assembly	Roche	http://www.broadinstitute.org/science/programs/genome-biology/crd
CAP3	Alignment/assembly	Roche	http://pbil.univ-lyon1.fr/cap3.php
Consensus/SeqCons	Alignment/assembly	Roche	http://www.seqan.de/downloads/projects.html
Curtain	Alignment/assembly	Illumina/Roche/ABI	http://code.google.com/p/curtain/
Edena	Alignment/assembly	Illumina	http://www.genomic.ch/edena
Euler-SR	Alignment/assembly	Illumina/Roche	http://euler-assembler.ucsd.edu/portal/?q=team
FuzzyPath	Alignment/assembly	Illumina/Roche	ftp://ftp.sanger.ac.uk/pub/zn1/fuzzypath/fuzzypath_v3.0.tgz
IDBA	Alignment/assembly	Illumina	http://www.cs.hku.hk/walse/idba/
MIRA/MIRA3	Alignment/assembly	Illumina/Roche	http://chevreux.org/projects_mira.html
Newbler	Alignment/assembly	Roche	roche-applied-science.com/
Phrap	Alignment/assembly	Illumina/Roche	http://www.phrap.org/consed/consed.html#howToGet
RGA	Alignment/assembly	Illumina	http://rga.cgrb.oregonstate.edu/
QSRA	Alignment/assembly	Illumina	http://qsra.cgrb.oregonstate.edu/
SHARCGS	Alignment/assembly	Illumina	http://sharcgs.molgen.mpg.de/
SHORTY	Alignment/assembly	ABI	http://www.cs.sunysb.edu/wskiena/shorty/
SHRAP	Alignment/assembly	Roche Illumina	By request http://soap.genomics.org.cn
SOAPdenovo	Alignment/assembly	Illumina/ABI	http://www.physics.rutgers.edu/%7Eanirvans/SOPRA/
SOPRA	Alignment/assembly	Roche	http://bioserver.cs.put.poznan.pl/sr-asm-short-reads-assembly-algorithm
SR-ASM	Alignment/assembly	Illumina/Roche	http://www.bcgsc.ca/platform/bioinfo/software/ssake
SSAKE	Alignment/assembly	Illumina	http://sourceforge.net/projects/taipan/files/
Taipan	Alignment/assembly	Illumina/Roche	http://sourceforge.net/projects/vcake
VCAKE	Alignment/assembly	Illumina/Roche/ABI	http://www.ebi.ac.uk/%7Ezerbino/velvet
Velvet	Alignment/assembly		
Reference-based assembly			
BFAST	Alignment/assembly	Illumina/ABI	http://sourceforge.net/apps/mediawiki/bfast/index.php?title=Main_Page

Bowtie	Alignment/assembly	Illumina/Roche/ABI	http://bowtie-bio.sourceforge.net
BWA CoronaLite	Alignment/assembly	Illumina/ABI	http://bio-bwa.sourceforge.net/bwa.shtml
CABOG	Alignment/assembly	ABI	http://solidsoftwaretools.com/gf/project/corona/
ELAND/ELAND2	Alignment/assembly	Roche/ABI	http://wgs-assembler.sf.net
EULER	Alignment/assembly	Illumina/ABI	http://www.illumina.com/
Exonerate	Alignment/assembly	Illumina	http://euler-assembler.ucsd.edu/portal/
EMBF	Alignment/assembly	Roche	http://www.ebi.ac.uk/wguy/exonerate
GenomeMapper	Alignment/assembly	Illumina	http://www.biomedcentral.com/1471-2105/10?issue=S1
GMAP	Alignment/assembly	Illumina	http://1001genomes.org/downloads/genomemapper.html
gnumap	Alignment/assembly	Illumina	http://www.gene.com/share/gmap
ICON	Alignment/assembly	Illumina	http://dna.cs.byu.edu/gnumap/
Karma	Alignment/assembly	Illumina	http://icorn.sourceforge.net/
LAST	Alignment/assembly	Illumina/ABI	http://www.sph.umich.edu/csg/pha/karma/
LOCAS	Alignment/assembly	Illumina	http://last.cbrc.jp/
Mapreads	Alignment/assembly	Illumina	http://www-ab.informatik.uni-tuebingen.de/software/locas
MAQ	Alignment/assembly	ABI	http://solidsoftwaretools.com/gf/project/mapreads/
MOM	Alignment/assembly	Illumina/ABI	http://maq.sourceforge.net
Mosaik	Alignment/assembly	Illumina	http://mom.csbc.vcu.edu/
mrFAST/mrsFAST	Alignment/assembly	Illumina/Roche/ABI	http://bioinformatics.bc.edu/marthlab/Mosaik
MUMer	Alignment/assembly	Illumina	http://mrfast.sourceforge.net/
Nexalign	Alignment/assembly	ABI Illumina	http://mummer.sourceforge.net/
Novocraft	Alignment/assembly	Illumina	http://genome.gsc.riken.jp/osc/english/dataresource/
PerM	Alignment/assembly	Illumina/ABI	http://www.novocraft.com/
RazerS	Alignment/assembly	Illumina/ABI	http://code.google.com/p/perm/
RMAP	Alignment/assembly	Illumina	http://www.seqan.de/projects/razers.html
segemehl	Alignment/assembly	Illumina/Roche	http://rulai.cshl.edu/rmap
SeqCons	Alignment/assembly	Roche	http://www.bioinf.uni-leipzig.de/Software/segemehl/
SeqMap	Alignment/assembly	Illumina	http://www.seqan.de/projects/seqcons.html
SHRiMP	Alignment/assembly	Illumina/Roche/ABI	http://biogibbs.stanford.edu/~jiangh/SeqMap/
Slider/SliderII	Alignment/assembly	Illumina	http://compbio.cs.toronto.edu/shrimp
SOCS	Alignment/assembly	ABI	http://www.bcgsc.ca/platform/bioinfo/software/slider
SOAP/SOAP2	Alignment/assembly	Illumina/ABI	http://solidsoftwaretools.com/gf/project/socs/
	Alignment/assembly		http://soap.genomics.org.cn

(continued on next page)

Table 3 (continued)

Program	Function	Platform	Website
SSAHA/SSAHA2	Alignment/assembly	Illumina/Roche	http://www.sanger.ac.uk/Software/analysis/SSAHA2
Stampy	Alignment/assembly	Illumina	http://www.well.ox.ac.uk/wmarting/
SXOligoSearch	Alignment/assembly	Illumina	http://synasite.mgrc.com.my:8080/sxog/NewSXOligoSearch.php
SHORE	Alignment/assembly	Illumina	http://1001genomes.org/downloads/shore.html
Vmatch	Alignment/assembly	Illumina	http://www.vmatch.de/
Diagnostics/utilities			
Artemis/ACT	Visualization tool	Illumina/Roche	http://www.sanger.ac.uk/resources/software/artemis/
CASHX	Pipeline	Illumina	http://seqanswers.com/wiki/CASHX
Consed	Visualization tool	Illumina/Roche	http://www.genome.washington.edu/consed/consed.html
EagleView	Visualization tool	Illumina/Roche	http://bioinformatics.bc.edu/marthlab/EagleView
FastQC	Quality assessment	Illumina/ABI	http://www.bioinformatics.bbsrc.ac.uk/projects/fastqc/
Gambit	Visualization tool	Illumina/Roche	http://bioinformatics.bc.edu/marthlab/Gambit
Goby	Data management	Illumina/Roche/ABI	http://campagnelab.org/software/goby/
G-SQZ	Data management	Illumina/ABI	http://public.tgen.org/sqz
Hawkeye	Visualization tool	Illumina/Roche	http://amos.sourceforge.net/hawkeye
Hybrid-SHREC	Error Correction	Illumina/Roche/ABI	http://www.cs.helsinki.fi/u/lsalmel/hybrid-shrec/
IGV	Visualization tool	Illumina	http://www.broadinstitute.org/igv/?q14home
LookSeq	Visualization tool	Illumina/Roche	http://lookseq.sourceforge.net
MagicViewer	Visualization tool	Illumina	http://bioinformatics.zj.cn/magicviewer/
MapView	Visualization tool	Illumina	http://evolution.sysu.edu.cn/mapview/
NGSView	Visualization tool	Illumina/ABI	http://ngsview.sourceforge.net
PIQA	Quality assessment	Illumina	http://bioinfo.uh.edu/PIQA
Reconciliation	Assembly pipeline	Illumina	http://www.genome.umd.edu/software.htm
RefCov	Sequence coverage	Illumina/Roche	http://genome.wustl.edu/tools/cancer-genomics
SAM Tools	Utilities	Illumina/Roche	http://sourceforge.net/projects/samtools/files/
Savant	Visualization tool	Illumina/Roche	http://compbio.cs.toronto.edu/savant/
ShortRead	Quality assessment	Illumina/Roche	http://bioconductor.org/packages/2.6/bioc/html/ShortRead.html
SHREC	Error Correction	Illumina/Roche	http://www.informatik.uni-kiel.de/jasc/Shrec/
Staden Tools (GAP5)	Pipeline	Illumina/Roche	http://sourceforge.net/projects/staden/files/
Tablet	Visualization tool	Illumina/Roche	http://bioinf.scri.ac.uk/tablet
TagDust	Data cleaning	Illumina	http://genome.gsc.riken.jp/osc/english/software/
TileQC	Quality assessment	Illumina	http://www.science.oregonstate.edu/wdolanp/tileqc
XMatchView	Visualization tool	Illumina/Roche	http://www.bcgsc.ca/platform/bioinfo/software/xmatchview
Yenta	Visualization tool	Illumina	http://genome.wustl.edu/tools/cancer-genomics
Geneus	Data management	Illumina/ABI	http://www.genologics.com/solutions/research-informatics/

The genetic variant prediction and detection bioinformatic programs for NGS data analysis.

Variant prediction/detection	Platform	Website
Functional variant prediction		
B-SIFT		http://research-pub.gene.com/bsift/
MAPP		http://mendel.stanford.edu/supplementarydata/stone_MAPP_2005
PhD-SNP		http://gpcr.biocomp.unibo.it/wemidio/PhD-SNP/PhD-SNP
PolyPhen-2/PolyPhen		http://genetics.bwh.harvard.edu/pph2/
SIFT SNAP		http://blocks.fhcrc.org/sift/SIFT.html
SNAPper/Pedant		http://www.rostlab.org/services/SNAP http://pedant.gsf.de/snapper
Variant detection		
Structural/genomic variant		
BreakDancer	Roche/Illumina/ABI	http://genome.wustl.edu/tools/cancer-genomics/
BreakDancer/BD- Mini	Roche/Illumina/ABI	http://seqanswers.com/wiki/BreakDancer
Breakway	Roche/Illumina/ABI	http://sourceforge.net/projects/breakway/files/
CNVSeq	Roche	http://tiger.dbs.nus.edu.sg/CNV-seq/
cnvHMM	Illumina	http://genome.wustl.edu/pub/software/cancer-genomics/cnvHMM/
cnD	Illumina	http://www.sanger.ac.uk/resources/software/cnd.html
GASV/GSV	Illumina	http://cs.brown.edu/people/braphael/software.html
Hydra	Illumina	http://code.google.com/p/hydra-sv/
MoDIL	Illumina	http://compbio.cs.toronto.edu/modil/
mrFAST	Illumina	http://mrfast.sourceforge.net/
NovelSeq	Roche/Illumina/ABI	http://compbio.cs.sfu.ca/strvar.htm
PEMer	Roche/Illumina/ABI	http://sv.gersteinlab.org/pemer/
Pindel	Illumina	http://www.ebi.ac.uk/wkyp/pindel/
SegSeq	Illumina/ABI	http://www.broadinstitute.org/
SOAPsv	Roche/Illumina/ABI	http://soap.genomics.org.cn
Solid large Indel tool	ABI	http://solidsoftwaretools.com/gf/project/large_indel/
Solid CNV tool SWT	ABI	http://solidsoftwaretools.com/gf/project/cnv/
VariationHunter/VH-CR	Illumina	http://genome.wustl.edu/pub/software/cancer-genomics/GSTAT/
VARIID	Illumina ABI	http://compbio.cs.sfu.ca/strvar.html http://compbio.cs.utoronto.ca/varid
Single nucleotide variant		
Atlas-SNP2	Roche/Illumina	http://www.hgsc.bcm.tmc.edu/cascade-tech-software-ti.hgsc
BOAT	Illumina	http://boat.cbi.pku.edu.cn/
DNA Baser	Roche	http://www.dnabaser.com/help/manual.html
DNA	Roche/Illumina/ABI	http://sourceforge.net/projects/dnaa/
Galign	Illumina	http://shahamlab.rockefeller.edu/galign/galign.htm
GigaBayes/PbShort	Roche/Illumina	http://bioinformatics.bc.edu/marthlab/GigaBayes
GSNAP inGAP	Roche/Illumina	http://share.gene.com/gmap
ngs_backbone	Roche/Illumina	http://sites.google.com/site/nextgengenomics/ingap
Omixon Variant	Roche/Illumina	http://bioinf.comav.upv.es/ngs_backbone/index.html
PyroBayes	ABI Roche	http://www.omixon.com/omixon/index.html
ssahaSNP	Illumina/Roche	http://bioinformatics.bc.edu/marthlab/PyroBayes
Slider	Illumina	http://www.sanger.ac.uk/Software/analysis/ssahaSNP
SNP-o-matic	Illumina	http://www.bcgsc.ca/platform/bioinfo/software/slider
SNPSeeker	Illumina	http://snpomatic.sourceforge.net
SNVMix	Illumina	http://www.genetics.wustl.edu/rmlab/
SOAPsnp	Roche/Illumina/ABI	http://compbio.bccrc.ca
SWA454	Roche	http://soap.genomics.org.cn
SVA	Illumina Illumina	http://www.broadinstitute.org/science/programs/genome-biology/crd
VAAL	Roche/Illumina	http://www.svapproject.org/
VarScan	Roche/Illumina/ABI	http://www.broadinstitute.org/science/programs/genome-biology/crd
VARIID		http://genome.wustl.edu/tools/cancer-genomics http://compbio.cs.utoronto.ca/varid
Differences between genomes		
DIAL	Illumina	http://www.bx.psu.edu/miller_lab/

SomaticCall
SWAP454
VAAL

Illumina
Roche
Illumina

<http://www.broadinstitute.org/science/programs/genome-biology/crd>
<http://www.broadinstitute.org/science/programs/genome-biology/crd>
<http://www.broadinstitute.org/science/programs/genome-biology/crd>

Source:

Jun Zhang , Rod Chiodini , Ahmed Badr , Genfa Zhang (2011) , The impact of next-generation sequencing on genomics Journal of Genetics and Genomics 38 (2011) 95–109