Open Reading Frame(ORF)

DNA (Deoxyribonucleic acid) is the genetic material that contains all the genetic information in a living organisms. The information is stored as genetic codes using adenine (A), guanine (G), cytosine(C) and thymine (T). During the transcription process, DNA is transcribed to mRNA. Each of these base pairs will bond with a sugar and phosphate molecule to form a nucleotide. Three nucleotides that codes for a particular amino acid during translation is called as a codon. The region of a nucleotide that starts from an initiation codon and ends with a stop codon is called an **Open Reading Frame(ORF)**. Proteins are formed from ORF. By analyzing the ORF we can predict the possible amino acids that might be produced during translation. The ORF finder is a program available at NCBI website. It identifies all ORF or possible protein coding region from six different reading frame.

DNA (Deoxyribonucleic acid) is the genetic material that contains the genetic information for development and helps in maintaining all the functions in a living organisms.The information is stored as genetic codes using four different bases. They are adenine (A), guanine (G), cytosine(C) and thymine (T). In two strands of DNA, adenine always pair with thymine and guanine pair with cytosine. Each of these base pairs will bond with a sugar and phosphate molecule to form a nucleotide. The base pairing of DNA will result in a ladder shape structure of these strands which is called a double helix. RNA is differs from DNA only in 1 base pair i.e. in RNA it is uracil (U) instead of thymine(T). mRNA (messenger RNA) is a type of RNA which is formed from DNA transcription. During the transcription process, DNA is transcribed to mRNA in the nucleus and moves to the cytoplasm through the nuclear pores. This mRNA is translated to protein in the cytoplasm with the help of ribosomes. In mRNA, 3 nucleotides are considered at a time since a set of 3 nucleaotides (refered to as codon) codes for an amino acid. The region of a nucleotide that starts from an initiation codon and ends with a stop codon is called an Open Reading Frame(ORF). An initiation codon is the triplet codon that codes for the first amino acid in the translation process. The translation process will start only with the initiation codon, ATG which codes for the amino acid methionine. The translation process stops when it comes across a stop codon. There are three stop codons: TAA ("ochre"), TAG ("amber") and TGA ("opal" or "umber"). Any of these codons can stop the translation. Genetic codon can form 64 triplets(4^3) from the 4 nucleotides that codes for amino acids. Protein is formed from the ORF.

How to find ORF

1. Consider a hypothetical sequence:
CGCTACGTCTTACGCTGGAGCTCTCATGGATCGGTTCGGTAGGGCTCGATCACATCGCTAGCCAT

2. Divide the sequence into 6 different reading frames(+1, +2, +3, -1, -2 and -3). The first reading frame is obtained by considering the sequence in words of 3.

FRAME +1: CGC TAC GTC TTA CGC TGG AGC TCT CAT GGA TCG GTT CGG TAG GGC TCG ATC ACA TCG CTA GCC AT

The second reading frame is formed after leaving the first nucleotide and then grouping the sequence into words of 3 nucleotides

FRAME +2: C GCT ACG TCT TAC GCT GGA GCT CTC ATG GAT CGG TTC GGT AGG GCT CGA TCA CAT CGC TAG CCA T

The third reading frame is formed after leaving the first 2 nucleotides and then grouping the sequence into words of 3 nucleotides

FRAME +3: CG CTA CGT CTT ACG CTG GAG CTC TCA TGG ATC GGT TCG GTA GGG CTC GAT CAC ATC GCT AGC CAT

The other 3 reading frames can be found only after finding the reverse complement.
Complement : **GCGATGCAGAATGCGACCTCGAGAGTACCTAGCCAAGCCATCCCGAGCTAGTGTAGCGATCGGTA**
Reverse
complement: **ATGGCTAGCGATGTGATCGAGCCCTACCGAACCGATCCATGAGAGCTCCAGCGTAAGACGTAGCG**

Now same process as that of +1, +2 and +3 strands is repeated for -1, -2 and -3 strands with reverse complement sequence

FRAME -1: ATG GCT AGC GAT GTG ATC GAG CCC TAC CGA ACC GAT CCA TGA GAG CTC CAG CGT AAG ACG TAG CG

FRAME -2: A TGG CTA GCG ATG TGA TCG AGC CCT ACC GAA CCG ATC CAT GAG AGC TCC AGC GTA AGA CGT AGC G

FRAME -3: AT GGC TAG CGA TGT GAT CGA GCC CTA CCG AAC CGA TCC ATG AGA GCT CCA GCG TAA GAC GTA GCG

3. Now mark the start codon and stop codons in the reading frames

FRAME +1: CGC TAC GTC TTA CGC TGG AGC TCT CAT GGA TCG GTT CGG TAG GGC TCG ATC ACA TCG CTA GCC AT

FRAME +2: C GCT ACG TCT TAC GCT GGA GCT CTC ATG GAT CGG TTC GGT AGG GCT CGA TCA CAT CGC TAG CCA T

FRAME +3: CG CTA CGT CTT ACG CTG GAG CTC TCA TGG ATC GGT TCG GTA GGG CTC GAT CAC ATC GCT AGC CAT

FRAME -1: ATG GCT AGC GAT GTG ATC GAG CCC TAC CGA ACC GAT CCA TGA GAG CTC CAG CGT AAG ACG TAG CG

FRAME -2: A TGG CTA GCG ATG TGA TCG AGC CCT ACC GAA CCG ATC CAT GAG AGC TCC AGC GTA AGA CGT AGC G

FRAME -3: AT GGC TAG CGA TGT GAT CGA GCC CTA CCG AAC CGA TCC ATG AGA GCT CCA GCG TAA GAC GTA GCG

4. Identify the open reading frame (ORF) - sequence stretch begining with a start codon and ending in a stop codon.

FRAME +2: ATG GAT CGG TTC GGT AGG GCT CGA TCA CAT CGC TAG

FRAME -1: ATG GCT AGC GAT GTG ATC GAG CCC TAC CGA ACC GAT CCA TGA

FRAME -3: ATG AGA GCT CCA GCG TAA

5. Based on the amino acid table the peptide sequence is found

First Nucleotide	Second Nucleotide: U code	U Amino acid	C code	C Amino acid	A code	A Amino acid	G code	G Amino acid	Third Nucleotide
U	UUU	phe	UCU	ser	UAU	tyr	UGU	cys	U
	UUC		UCC		UAC		UGC		C
	UUA	leu	UCA		UAA	STOP	UGA	STOP	A
	UUG		UCG		UAG	STOP	UGG	trp	G
C	CUU	leu	CCU	pro	CAU	his	CGU	arg	U
	CUC		CCC		CAA		CGC		C
	CUA		CCA		CAC	gln	CGA		A
	CUG		CCG		CAG		CGG		G
A	AUU	ile	ACU	thr	AAU	asn	AGU	ser	U
	AUC		ACC		AAC		AGC		C
	AUA		ACA		AAA	lys	AGA	arg	A
	AUG	met	ACG		AAG		AGG		G
G	GUU	val	GCU	ala	GAU	asp	GGU	glv	U
	GUC		GCC		GAC		GGC		C
	GUA		GCA		GAA	glu	GGA		A
	GUG		GCG		GAG		GGG		G

Figure 1: Amino Acid Table

FRAME +2: ATG GAT CGG TTC GGT AGG GCT CGA TCA CAT CGC TAG
met asp arg phe gly arg ala arg ser his arg stop

FRAME -1: ATG GCT AGC GAT GTG ATC GAG CCC TAC CGA ACC GAT CCA TGA
met ala ser asp val ile glu pro tyr arg thr asp pro stop

FRAME -3: ATG AGA GCT CCA GCG TAA
met arg ala pro ala stop

By analyzing the ORF we can predict the possible amino acids that are producing during the translation process. The prediction of the correct ORF from a newly sequenced gene is an important step. Finding ORF helps to design the primers which are required for experiments like PCR, sequencing etc.

ORF Finder:

The ORF finder is a program available at NCBI website. It identifies the all open reading frames or the possible protein coding region in sequence. It shows 6 horizontal bars corresponding to one of the possible reading frame. In each direction of the DNA there would be 3 possible reading frames. So total 6 possible reading frame (6 horizontal bars) would be there for every DNA sequence. The 6 possible reading frames are +1, +2, +3 and -1, -2 and -3 in the reverse strand. The resultant amino acids can be saved and search against various protein databases using blast for finding similar sequences or amino acids. The result displays the possible protein sequence and the length of the open reading frame etc.

FOR BLAST “REFER MODULE 5 NOTE”

Sequence assembly

K. Scheibye-Alsing[1], S. Hoffmann[2], A. Frankel, P. Jensen, P. F. Stadler[2,6,3,4,5],
Y. Mang[7], N. Tommerup[7]
M. J. Gilchrist [8] A.-B. Nygård,
S. Cirera[1], C. B. Jørgensen[1], M. Fredholm[1] and J. Gorodkin[1,‡]

[1]Division of Genetics and Bioinformatics, IBHV, University of Copenhagen,
Grønnegårdsvej 3, 1870 Frederiksberg C, Denmark
[2] Interdisciplinary Center for Bioinformatics, University of Leipzig,
Härtelstraße 16-18, D-04107 Leipzig, Germany
[3] Bioinformatics Group, Dept. of Computer Science,
University of Leipzig, Härtelstraße 16-18, D-04107 Leipzig, Germany
[4]RNomics Group, Fraunhofer Institut für Zelltherapie und Immunologie,
Deutscher Platz 5e, D-04103 Leipzig, Germany
[5]Santa Fe Institute, 1399 Hyde Park Rd., Santa Fe, NM 87501, USA
[6]Department of Theoretical Chemistry, University of Vienna,
Währingerstraße 17, A-1090 Wien, Austria
[7] Wilhelm Johannsen Centre for Functional Genome Research,
Department of Cellular and Molecular Medicine, Panum Institute, University of Copenhagen,
Blegdamsvej 3B, DK-2200 Copenhagen N, Denmark
[8] The Wellcome Trust/Cancer Research UK Gurdon Institute
Cambridge CB2 1QN

Draft: April 27, 2009

‡ Corresponding author: Jan Gorodkin
Division of Genetics and Bioinformatics, IBHV
University of Copenhagen
Grønnegårdsvej 3
1870 Frederiksberg C
Denmark
Phone: +45 3533 3578
Fax: +45 3533 3042
Email: gorodkin@genome.ku.dk

Abstract:
Despite the rapidly increasing number of sequenced and re-sequenced genomes, many issues regarding the computational assembly of large-scale sequencing data have remain unresolved. Computational assembly is crucial in large genome projects as well for the evolving high-throughput technologies and plays an important role in processing the information generated by these methods. Here, we provide a comprehensive overview of the current publicly available sequence assembly programs. We describe the basic principles of computational assembly along with the main concerns, such as repetitive sequences in genomic DNA, highly expressed genes and alternative transcripts in EST sequences. We summarize existing comparisons of different assemblers and provide a detailed descriptions and directions for download of assembly programs at: `http://genome.ku.dk/resources/assembly/methods.html`.

Keywords: Assembly methods, EST, shotgun, genomes, high-throughput sequencing.

1 Introduction

Genome sequencing is a discipline that has undergone tremendous development in the past. With the introduction of the different new massively parallel sequencing technologies the field will go through further transformations as new challenges arise. Today 567 bacterial genomes with up to 10.5 million base pairs (*Plesiocystis pacifica SIR-1*) have been sequenced and submitted to NCBI (as of October 9, 2008). In addition several eukaryote genomes with approximately three billion base pairs have been sequenced and assembled (`http://www.ensembl.org`), and many other sequencing projects are under way (`http://www.genomesonline.org`) [1].

The experimental technique used in most de novo sequencing projects of higher organisms, DNA chain termination, was developed three decades ago and remains, except for much higher levels of automation, basically the same. The introduction of new massively parallel sequencing methods, however, opens completely new fields of application. Shortly after the introduction of sequencing methods, some of the first reports of the determination and comparison of cDNA sequences were published. Late in the 1970s the bacteriophages phiX174 and Lambda [2, 3, 4] were among the first genomes to be completed together with the human mitochondrion [5, 6].

In the following decade the shotgun sequencing strategy was introduced [7, 8], and during the subsequent years it was extended by applying it to larger and larger DNA sequences cloned in plasmids (a few kilobases (kb)), cosmids (40 kb) [9], artificial chromosomes cloned in bacteria (BAC – Bacterial Artificial Chromosome) and yeast (YAC – Yeast Artificial Chromosome), with inserts of 100 to 500 kb [10]. The assembly of whole genome shotgun sequencing data was deemed to be futile until the successful WGS assembly of the 1.8Mb genome *Haemophilus influenzae* in 1994 [11]. An approximate time line of the major breakthroughs and milestones in sequencing is shown on Fig. 1.

[Figure 1]

“High throughput” sequencing (HTS) of cDNA was initiated in 1991 by Adams [12], who also introduced the term "Expressed Sequence Tag" (EST) to refer to this new type of sequence information. Collections of ESTs have given a first good approximation of the diversity of all protein coding genes in a tissue [13]. During the years ESTs have become an important tool with many applications, mostly in relation to gene analysis and gene discovery [14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38]

The amount of data generated by the different sequencing projects is overwhelming. For example, sequencing of the human genome produced 23 and 27 billion bases of raw shotgun sequences in the International Human Genome Sequencing Consortium and the Celera projects, respectively [39, 40]. However, the vast amount of fragments can not readily be concatenated to a final sequence. Only by using computers it becomes possible to carry out the assembly of the pieces, but the outcome as well as the reliability of the result for a given type of data depends on the underlying strategy implemented in the computer program. Some strategies might be more suited for one type of data than others. Also, the computational resources of some methods might not scale well with the number of sequences in the data set. Though the experimental techniques have essentially driven the computational aspect of sequence assembly, the computational aspect is still of utmost importance since any meaningful assembly needs to be computer assisted.

One of the first assemblers introduced by Staden in 1980 [41] was a computer program developed to store and manipulate DNA gel reading data obtained from the shotgun method of DNA sequencing. During the next decade several other programs were presented, among them SEQAID [42], CAP [43], PHRAP [44], and the TIGR assembler, which was used to assemble the genome of *Haemophilus influenzae* [11]. In order to assemble larger and more complex eukaryotic genomes, new assemblers have been designed and implemented. Among them the Celera Assembler (now part of AMOS) [45, 46] and GigAssembler [39], both applied to human genome data sets; the JAZZ-assembler, which was applied to both the genome of *Takifugu rubripes* (the pufferfish) [47] and *Ciona intestinalis* [48]; and the ARACHNE [49] and Phusion [50] assemblers, both applied to the mouse genome.

Several specific efforts have been undertaken in the context of EST assembly, and several tools are available. Among them are StackPack [51, 52], TIGR TGICL [53], and geneDistiller [54]. Some of the tools deal with splice variants [55] or other problems such as chimerism (and includes alternative splice variants detection) [54, 56, 57]. Approaches to incorporate rather than remove repetitive sequences are discussed in [58, 59, 60].

Along with the increasing number of completed genomes, efforts are also made in developing computational methods for comparing genomes. These include TIGRs MUMmer [61, 62], TWINSCAN, GENEWISE, GENOMESCAN [63, 64],

BLAT [65], and AVID [66, 67] used for alignment and comparison of whole genomes, and FORRepeats which is used to detect repeats on entire chromosomes and between genomes [68].

The massive effort to sequence the human genome produced a first draft version in 2001 [39], and did, as a draft sequence, contain numerous gaps. It took another 3 years of sequencing and assembly before the finished version was presented (which still contains more than 300 gaps) [69].

2 Sequencing approaches

As mentioned the choice of assembly strategy depends on the sequencing method, and the choice of sequencing method may also depend on the organism that is being sequenced. Issues that can affect the final assembly (other than the obvious quality of sequence data) are the size of the inserts, whether the sequencing was uni- or bi-directional, the library construction, the cloning vector, the selection of clones to be sequenced, and the availability of additional information (consensus genome, ESTs, known verified genes, gene maps, etc.).

Approaches for the de novo sequencing of genomes from higher organisms using Sanger sequencing [70] will be described first. In the context of genome resequencing we take a closer look on the new massively parallel sequencing technologies and their obstacles, though many of the concerns are overlapping *eg.* sequencing quality assessment.

2.1 Basic sequencing procedure

The basic procedure in sequencing has been to isolate genomic DNA or RNA (reverse transcribed into cDNA), and clone it into vectors (*eg.* plasmids, BACs) capable of stable propagation in suitable host cells such as *Escherichia coli*, see Fig. 2 for a schematic illustration of a sequencing vector. Several cloning systems with insert sizes varying from hundreds of base pairs to megabases have been developed. The ideal clone library for genomic sequencing has the following features.

1. The clones are highly redundant, covering the entire genome many times (typically 6–10).
2. The clone coverage is random and not biased towards or against specific regions of the genome.
3. The clones are stable, not subject to recombination or reorganization during the propagation process [71].

It should be noted that one of the major improvements of the new massively parallel sequencing technologies is that they do not rely on vector cloning prior to sequencing, and the concerns listed here are therefore not directly applicable to those technologies.

[Figure 2]

After propagation, the clones are selected and the sequencing is performed. An essential feature in sequencing is the attachment of quality values to the raw sequences. The quality values indicate the likelihood of each base call being correct. In the assembly stage the quality values will help to distinguish true DNA polymorphisms from sequencing errors and match end sequences of low quality [72, 73, 74, 75].

In genomic shotgun sequencing, which typically uses a single individual DNA source, sequences sharing less than 98% identity are usually assumed to come from different regions of a genome (including different repetitive elements) [76]. In contrast, EST data is usually derived from a variety of sources representing the spectrum of polymorphisms in the original samples. These will usually include a number of erroneous polymorphism which are caused by sequencing errors inherent in single pass sequencing, a relatively high rate of insertions and deletions, contamination by vector and linker sequences and the non-random distribution of sequence start sites in oligo(dT)-primed libraries. Therefore, the degree of identity in overlapping sequences from the same gene will often be lower in EST projects than in genomic sequencing projects. In addition, the patterns of overlapping sequences caused by alternative spliceforms are different from those observed in a genomic shotgun project [76].

The major tool to gather sequence information was the method introduced by Fred Sanger in the second half of the 70'ties. It uses dideoxynucleotide triphosphates (ddNTPs) as DNA chain terminators [77, 70]. The classical Sanger approach is carried out in four independent DNA polymerase reactions. Besides the DNA template and deoxynucleotides (dNTPs) a reaction mix contains either ddATP, ddCTP, ddTTP or ddGTP. Each reaction results in DNA fragments of different length terminating with the respective ddNTP. Electrophoresis of the fluorescence- or radio labeled fragments allows the recovery of the template sequence. Later, the use of dye-terminators made it possible to perform sequencing in a single reaction rather than four – the basic principle however remained the same. While the classical Sanger approach requires separate synthesis and detection steps, High Throughput Sequencing (HTS) technologies employ sequencing-by-synthesis and sequencing-by-ligation approaches, allowing for simultaneous synthesis and detection.

2.2 Shotgun sequencing

Two approaches for genome shotgun sequencing can be distinguished: whole-genome shotgun (WGS) sequencing and hierarchical shotgun sequencing.

2.2.1 Whole Genome Shotgun

Sequencing using the whole genome shotgun approach basically means that the genome is randomly broken into pieces and cloned into a sequencing vector. The inserts are subsequently processed to generate sequences of bases (referred to as reads). See illustration on Fig. 3a. During the mid 1990s several groups recognized that sequence information from both ends of relatively long inserts dramatically improves the efficiency of sequence assembly [9, 78, 79, 80, 81, 82]. In contrast to single sequence reads from one end of the shotgun clones pairs of sequence reads from both ends have known spacing and orientation. Exact knowledge of the length of the insert is not required to utilize the advantages of end sequencing in assembly [83], but good estimates of clone length will aid the assembly immensely.

[Figure 3]

2.2.2 Hierarchical shotgun sequencing

The 'Hierarchical shotgun sequencing' (also referred to as 'map-based', 'BAC-based' or 'clone-by-clone') approach involves generating and organizing a set of large insert clones (typically 100–200 kb each) covering the genome (a "minimal tiling path"), followed by separate shotgun sequencing on each clone. For illustration see Fig. 3. It is possible to establish a tiling path of overlapping BAC-clones using only BAC fingerprinting technologies [84]. However, knowledge of unique genome markers (*eg.* ESTs or sequence-tagged sites (STS)) and their location in the genome map is of great help for organizing the BAC clones in the correct order. In hierarchical shotgun sequencing the sequence information is local, therefore the risk of long-range and short-range misassembly is reduced.

2.2.3 Mixed strategy sequencing

A strategy that can be used on large complex genomes is the 'mixed strategy sequencing'. The technique utilizes both hierarchical and whole-genome shotgun. The method combines a light (x1) BAC clone coverage of the genome, with whole genome shotgun sequencing . The BAC clones act as a basic framework for WGS sequence assembly. The method was successfully applied to rat genome [85].

2.2.4 Reduced Representation Sequencing

A variant of WGS is "reduced representation sequencing" (RRS), where one selectively chooses subsets of the genome to avoid sequencing the (often much) larger regions that are not of interest. In [86], SNPs were discovered by mixing DNA from many individuals, preparing a library of appropriately sized restriction fragments, and randomly sequencing

clones. Here, the choice of the restriction fragments effectively selects only a small subset of the human genome. Several approaches to RRS have been employed for plant genomes [87, 88, 89]: Methyl-filtration (MF) sequences uses the endogenous restriction-modification system of *E. coli* to eliminate methylated DNA inserts, the RescueMu (RM) approach focuses on the gene-rich regions which are rich in mutator transposons, and High-Cot filtration avoids repetitive and low-copy sequences due to differences in the relative rates of DNA re-association. Most of the Maize and Sorghum genomes have been sequenced using MF.

Many of the applications of the new high throughput sequencing platforms are based on various RSS strategies (see 2.2.6). This includes electrophoretic size separation to enrich for small RNA molecules (*eg.* [90, 91]); reduced representation bisulphite sequencing for genome wide methylation analysis [92]; flow sorting of derivative translocation chromosomes for breakpoint mapping [93]; enrichment of DNA-fragments bound to specific proteins by chromatin immunoprecipitation of fixed, sheared DNA, for identification of transcription factor binding sites (CHiP-Seq) [94, 95, 96]; enrichment of specific parts of the genome by multiplex PCR-amplification [97] or by hybridization to custom made arrays [97, 98], *eg.* for SNP discovery [99] and in situ exon capture [100].

2.2.5 EST sequencing

Expressed Sequence Tags (ESTs) are sequences representing genes which can originate from specific tissues [12]. In EST-sequencing a single automated sequencing from one or both ends of a cDNA-inserts is performed. This single-pass approach is the major reason EST-sequencing is cost effective [101]. For additional information, see *eg.* [102] and references therein.

In most cases EST sequencing projects are aimed at establishing partial sequences of transcribed genes rather than full length cDNA sequences. However, this approach features some special challenges such as common sequence motifs, alternative transcripts and paralogous genes are challenges that potentially impact the assembly quality. These issues will be discussed further in section 4.4.3.

2.2.6 Massively parallel sequencing

Recently, a number of new sequencing technologies have emerged. The development was initiated by 454 sequencing and followed by Solexa sequencing and others [103, 104, 105, 106, 107]. The common feature of all these technologies is that they are massively parallel, *ie.* they generate a large number of different sequence reads in a single run. The generated small reads are usually aligned to a reference genome, and further analyzed, see Fig. 3d for an illustration.

The methods generally use one variant or another of fixing many sequence fragments on a substrate, cyclically adding different bases with some – technology-specific – luminal characteristics, and recording an image at each cycle. Image analysis is used to recover the all sequences at once. Sequencing of all immobilized fragments thus proceeds in parallel.

Compared to traditional sequencing a large amount of sequence data is generated at a drastically reduced cost per base. The most important disadvantage of high throughput sequencing is the significantly reduced read length, which limits their application in de novo sequencing of complex genomes (*eg.* due to repeats), at least using simple shotgun strategies. However, these new platforms have many uses in genome resequencing, especially if it is possible to align the fragments to an existing good quality reference genome.

Due to the amount of raw sequence data, high throughput sequencing is valuable in areas such as SNP finding. In EST sequencing, HTS technologies might enable a researcher to make accurate digital expression profiles, even including low abundance transcripts, and help detecting alternative splicing (depending on the platform chosen).

One of the key technologies that gave rise to the era of HTS, pyrosequencing, was introduced in 1998 [103]. This sequencing-by-synthesis method is at the very heart of GS FLX systems by 454 Life Sciences [104]. The detection is based on pyrophosphates (PPi) released during the polymerase reaction. Sulfurylase converts PPi to ATP which is subsequently consumed by luciferase to emit light in the visible spectrum. In GS FLX systems, a library of DNA templates is immobilized on DNA capture beads, amplified using emulsion PCR (emPCR) and loaded onto proprietary titer plates with several hundreds of thousands reaction wells. During a run, the four nucleotides are flowed sequentially over the

plates. The luciferase reaction triggered by nucleotides complementary to DNA templates is recorded by a CCD camera. A washing step is necessary to allow the next detection step. The GS FLX currently allows read lengths of several hundred bases. According to the manufacturer a single instrument run with two high-density plates generates information for about 20 million base pairs.

A competing technology, Solexa, now sold by Illumina (`http://www.illumina.com`), uses optically transparent surfaces to immobilize fragmented and adapter-tagged DNA. Each attached fragment is subsequently amplified $\sim$1000 fold by repeated steps of bridge amplification. The resulting clonal clusters are then sequenced using reversible terminators with removable fluorescent dyes. With approximately 30-40 bp Solexa reads are significantly shorter compared to GS FLX. However, close to 50 million clones per flow cell can be sequenced in parallel, resulting in presently >1.5 Gb of sequenced DNA in a single sequencing run. This amount can be doubled by sequencing the other end of each fragment (paired-end). Improvement in chemistry may further increase the read lengths and hence push the total amount of sequenced DNA well beyond the size of a human diploid genome.

A third synthesis-based technology, tSMS (true Single Molecule Sequencing), is currently distributed by Helicos (`http://www.helicosbio.com`). No DNA amplification is required for this approach. Instead, fragmented single stranded DNA molecules are directly immobilized on a solid surface. Similar to Solexa, tSMS works with nucleotides that carry a removable, laser light-detectable fluorescent. At the moment the system is able to sequence reads with lengths up to 55 bases at a speed of 25 to 90 million usable bases per hour.

SOLiD, a system now sold by Applied Biosystems (`http://www.appliedbiosystems.com`), is a technology that uses a sequencing-by-ligation approach. An adapter-tagged library of short DNA fragments is amplified with emPCR, immobilized on capture beads and then deposited onto high-density glass arrays. The SOLiD sequencing-by-ligation protocol uses four by four sets of 8-mer probes. In each set only two bases, fluorescently labeled, are specific. The interrogation of sequences is done in four phases. If a probe has specifically bound to the free template in the first phase, say at position 1 and 2, it is enzymatically ligated to the current 5′ end at position 0. After the detection step 3 nucleotides of the probe along with the fluorescence label are cleaved. The next ligation step interrogates 6 and 7 and so forth. After the first phase, the ligated sequence is removed, and another set of bases are called. So in the second phase bases 2 and 3 are read, in the third 3 and 4 and so forth. The advantage of the SOLiD system is that the double base reading leads to an increased accuracy. Currently SOLiD produces read lengths of about 30-40 bp and a total of 9 Gb per single run, with read length expected to become longer in the future.

In the future, other technologies may become available, such as the use of solid state nanopores for sequencing of single DNA molecules [108]. We refer to [109] for an overview, in which several interesting ideas how this approach could be implemented in practice are presented.

3 Mapping of short high-throughput sequencer reads

Compared to de novo assembly, the mapping of resequenced reads to a template genome is a computationally easier problem. Still, efficient mapping tools are crucial (see section 4.7), and several tools for mapping of short reads are available. Most of the tools, *ie.* MAQ, SOAP, SHRiMP or Eland (proprietary), use seeding techniques that gain their speed from precomputed hash look-up tables [110, 111, 112]. Typically, seeds of fixed length allow for not more than one or two mismatches. In addition, the capability to detect insertions and deletions, as they frequently occur in 454 sequences (see section 4.4.4) is very limited,and most programs can only detect indels in subsequent alignment runs. For short sequences it would be helpful, but computationally more expensive, to incorporate indels right from the start. Current mapping tools have different additional features. The program MAQ, *eg.*, additionally supports paired-end read matching — helpful to deal with paired-end reads produced *eg.* by the GS FLX and other high throughput platforms.

4 Computational assembly

Computational assembly is the only way to efficiently assemble sequenced fragments of DNA. However, a sufficient amount of high quality sequences are required. The assembly programs should be able to handle large data sets effectively

and avoid misassemblies in the presence of large repetitive or duplicated regions and redundant sequences. To accomplish this, effective algorithms to handle large input data sets with the use of minimal computer time and memory are needed.

One of the primary difficulties in computational genome assembly is to develop an algorithmic approach capable of detecting stretches of repetitive DNA without causing misassemblies. Repetitive sequences complicate assembly as different pieces of sequence can share the same repeat sequence originating from different genomic locations. Since the pieces are put together by searching for matching overlapping nucleotides, repeats can be put together erroneously. Typically, for shotgun data, repetitive sequences are revealed by clusters containing more overlapping reads than would be expected by chance, illustrated on Fig. 4.

[Figure 4]

In EST datasets the main difficulty is to develop an algorithmic approach that, in addition to efficient assembly, can handle highly expressed genes, paralogous genes, alternative spliceforms and chimerism in the dataset.

The theoretical background for genome assembly lies in computer science, and an insight into the mathematical and theoretical background can be found in [113] and references therein.

Although pyrosequencing with a whole-genome shotgun approach has been successfully applied to bacterial genomes [104], the construction of high-quality assemblies with high-throughput sequencing data is still a non-trivial problem even for short genomes. At present, no approach has been proposed to directly assemble large animal or plant genomes directly from short sequences obtained using HTS. As described below the SHort Read Assembly Protocol (SHRAP) [114], however, comprises a protocol for high-throughput short read sequencing that differs in two respects from classical hierarchical sequencing approaches. This protocol however, expects read lengths much longer (200 nucleotides) than those produced by SOLiD or Solexa. The assembly methodology is based on the Euler engine introduced in 2004 [60]. The Euler-SR assembler, specifically designed to assemble short reads, uses an updated version of the Euler engine to reduce memory requirements. The results for real Solexa reads, however, were less convincing [115] due to the poorly understood error model and highly variable error rates across different machines and run times.

4.1 Basic principles of Assembly

For the majority of traditional assembly programs the basic scheme is the same, namely the overlap-layout-consensus approach. Essentially it consists of the following steps [44, 116]:

- Sequence and quality data are read and the reads are cleaned.
- Overlaps are detected between reads. False overlaps, duplicate reads, chimeric reads and reads with self-matches (including repetitive sequences) are also identified and left out for further treatment.
- The reads are grouped to form a contig layout of the finished sequence.
- A multiple sequence alignment of the reads is performed, and a consensus sequence is constructed for each contig layout (often along with a computed quality value for each base).
- Possible sites of misassembly are identified by combining manual inspection with quality value validation.

Prior to the assembly, the electropherogram (for Sanger sequencing, images for massively parallel sequencers) for a given sequence is interpreted as a sequence of bases (a read) with associated quality values, these values reflect the log-odds score of the bases being correct. The basecaller PHRED [117] is often used, however alternatives exist, *eg*. the CATS basecaller [118].

The reads can then be screened for any contaminant DNA such as *Escherichia coli*, cloning or sequencing vector. Low quality regions can be identified and removed [45]. Base quality values can be used in computation of significant overlaps and in construction of the multiple alignments [44, 116]. The pipeline for a typical sequence assembly is sketched on Fig. 5.

[Figure 5]

For high-throughput sequencing data, the basic proposition for SHRAP is to sample clones from the genome at high coverage, while sequencing reads from these clones at low coverage. SHRAP starts off with assembling the reads greedily to small local assemblies and subsequently to contigs on each clone. It proceeds by ordering the clones in a "clone graph", and constructing "clone contigs", which are then assembled independently. Computer simulations of the procedure show that the approach can reach a quality comparable to the current assemblies of single human chromosomes and fruit fly genomes using reads of 200nt with an error rate of not more than 1%. These are constraints that are too strict for short (Solexa or SOLiD) reads ($\approx 40bp$) and because of higher error rates challenging for real 454 reads [119]. Furthermore, for mammalian genomes the use of a hierarchical sequencing strategy might be somewhat cumbersome.

However, the use of templates might bail Solexa and SOLiD users out: In a recent study, de novo assemblies of chloroplast genomes ($\approx$ 120 kb) were improved by aligning preassembled contigs to reference genomes [120]. After de-Bruijn graph assembly of reads [121], small contigs were aligned to closely related chloroplast genomes. Between 67% and 98% of the contigs could be aligned to such templates. If alignment failed, sequences were scanned for similarity using BLASTN [122]. The authors reported that successful BLASTN matches typically contained $>$ 100 bp insertions relative to the reference genome. In the end, however, their assemblies were estimated to be 88–94% complete. Yet, the assembly of mammalian genomes or genomes without good reference sequences seems to be a considerably more difficult task. The successful de novo assembly of Chloroplasts genomes with 454 reads has been shown earlier [123].

454, SOLiD, Solexa technologies allow convenient generation of mate-pair/paired-end sequences, *ie.* the ability to sequence both ends of each DNA fragment. However, in an assembly using a hybrid dataset of real 454 reads and simulated mate-pair data, about 96% of the mate-pairs did not contribute additional information and hence did not improve the assembly [115]. Likewise, in a hybrid dataset of 454 reads and Sanger reads the vast majority of long sequences did not improve the assembly substantially, measured by N50 contig size. Hence, the authors concluded that hybrid protocols should be reviewed critically. Despite those simulation results, the latter method has already been shown to work quite well in practice [124], and one area where mate-pair/paired-end sequencing should improve the analysis dramatically is for the detection of breakpoints related to structural rearrangements, *eg.* deletions, duplications, inversions and translocations [125].

4.2 General Assembler differences

When different assemblers try to piece the DNA puzzle together they essentially work from the same input, but the assemblers differ in the way they utilize the sequence information, and in the way this is combined with additional information. In general the differences fall in the following categories.

- **Overlaps:** A lot of different methods are used to find potential overlaps between sequences. Some are based on BLAST (*eg.* geneDistiller [54, 56]), while other assemblers use various other methods to find similarities between reads.
- **Additional information:** Depending on how the sequence reads are produced some additional information might be available. This information might consist of read pair information, BAC clone information, base quality information, etc. Some assemblers use this data to impose additional structure on the assembly of the sequences (*eg.* GigAssembler [39]).
- **Short read assembly:** De novo assembly of the micro reads generated from next generation sequencing platforms is still challenging. While assemblers have been developed and applied to assemble bacterial genomes successfully [115, 126], on larger genomes the assembly is performed by mapping the micro reads to reference genomes. The major next generation sequencing platforms all have built-in software to handle this task, *eg.* GS Reference mapper, Gerald for Solexa. In SOLiD systems the mapping tool "mapreads" converts reference sequences into color space and perform the mapping in color space.

A somewhat related issue is how the sequences are cleaned of contaminant sequences (*ie.* vector sequences, repeat sequences, *etc.*). While this can essentially be considered separately and independently from the assembly itself, some assemblers incorporate cleaning in the way they process the reads (*eg.* [49]).

These basic ideas will be discussed further in the following text, and an overview on how the different assemblers applies these ideas can be found in the supplemental material (`http://genome.ku.dk/resources/assembly/methods.html`).

4.3 Overlap identification and alignment

In a whole-genome context, trillions of overlaps between reads are examined [45]. The majority of assemblers uses alignment algorithms which are general modifications of methods first introduced by Needleman and Wunsch in 1970 [127], Smith and Waterman in 1981 [128] and Gotoh in 1982 [129].

Initial overlap detection is often performed by finding exact identical subsequences (often called words, k-words or k-mers) between reads, prior to making the actual alignments. These identical subsequences are used to find pairs of potentially overlapping sequences, which can then be aligned to each other in order to check if they represent a true overlap. The size of the subsequences varies from method to method, and is dynamic in some assemblers. Furthermore, the identical subsequences are grouped and used in different ways depending on the assembler.

For almost all assemblers, a modified Smith-Waterman [128] algorithm is used to align candidate overlapping reads. The basic idea in the alignment algorithms is to use dynamic programming to construct a matrix containing scores of all subsequences, which is then analyzed to find the "optimal" alignment. Dynamic programming simply means that the alignment is calculated as extensions to already aligned subsequences. The assembly programs differ in their exact implementation of this algorithm, as (nearly) all of them use a heuristic approach to decrease the computational load, thereby increasing speed (*eg.* [116]). In the assembly of ESTs a clustering step is used to group the input sequences sharing significant regions of near identity together [130]. On Fig. 6, an assembled cluster is shown, the example is taken from the Sino-Danish pig EST sequencing project [131].

[Figure 6]

4.3.1 Multiple alignments and the consensus sequence

While the alignment of two sequences is usually straightforward, aligning more than two is not so simple. The standard Smith-Waterman algorithm can easily be extended to the task of aligning many sequences by constructing a "multi-dimensional matrix". However, the number of calculations rise exponentially with the number of sequences. This sets severe practical limits of the number of sequences that are viable to align, and therefore finding the true sequence from a number of overlapping reads becomes difficult.

Precisely how the different assemblers generate a multiple alignment and consensus sequence is only vaguely described in the literature, but a common approach is to use a heuristic greedy algorithm (see for example [132]). The greedy algorithm typically performs pairwise alignment between overlapping reads, from which a multiple alignment is build up iteratively, *ie.* adding one sequence at a time, but with this approach there is no guarantee that such a multiple alignment is correct.

After the multiple alignment has been constructed the consensus sequence is found. This would typically be the sequence generated by taking the most common base at each position in the alignment, however other methods exist. For instance geneDistiller [54], where ungapped alignments of reads is performed (thus simplifying the multiple alignment). The consensus sequence is constructed by splitting the multiple alignment in 12-mer words and analyzing the relative frequencies of these, where the presence of alternative transcripts is detected through the frequencies of the 12-mers (and displayed as stretches of 'alternate consensus').

The assumption is that the final consensus sequence correspond to the original genomic sequence where the sequenced fragment originate.

4.3.2 Eulerian Fragment Assembly

In assemblers aimed at short read assembly (*eg.* SOLiD reads) an approach based on mathematical graph theory is often used, namely the Eulerian fragment assembly method. The Eulerian fragment assembly avoids the costly computation of pairwise alignments between reads [133]. The *De Bruijn graph* of a genome has as its vertices all distinct $k-1$ tuples that occur within the sequence (where k is the word length that is used). A directed edge is inserted between s and t if there is a k tuple $\langle u_1, u_2, \ldots, u_{k-1}, u_k \rangle$ in the genome such that $s = \langle u_1, u_2, \ldots, u_{k-1} \rangle$ and $\langle t = u_2, \ldots, u_{k-1}, u_k \rangle$, *ie.*, if s and t appear shifted by single nucleotide. A sketch of a graph construction procedure is shown on Fig. 7. In practice one uses the k-tuples appearing in the collection of the sequence reads and a value of k between 6 and 9 or 10. In the error-free case, the genomic sequence can be read off directly as an Eulerian path through the De Bruijn graph (with repeats forming "tangles"). In real, error-prone data underrepresented k-tuples, *ie.* k-tuples that appear less frequently than expected from the coverage rate, indicate sequencing errors and can be omitted.

[Figure 7]

4.4 Data reliability

4.4.1 Preprocessing and cleaning

A critical aspect of any large-scale sequencing effort is the production of high quality data. To obtain this preprocessing is applied to the reads. For Sanger sequencing this includes base-calling, filtering of low quality reads, short length reads (typically less than 100 bp), identification of sequence features such as linker restriction sites, cloning vectors, polyadenylation tails, library tags, polyadenylation signals [134] and other contaminants like bacterial sequences [135].

There are different computational programs available to detect these contaminations. Most of the existing programs used for processing solely focus on a single step. While PHRED [117] deals with base-calling, cross_match [44] aims to identify and mask vector sequences in reads. Preprocessing can also be done using other programs such as LUCY [135], a sequence trimming script like SeqClean [136], or ESTprep [134].

In the Solexa system, the module for sequence alignments, Gerald, applies some filters to remove low quality base calling before the real mapping starts. As it is based on optical detection of ultra-high dense sequence clusters on surface, chastity and purity of optical signals are crucial for accessing the quality. Distance between clusters is also taken into consideration. Thresholds for these features can be customized in the program (see Illumina in-house documentation for details). Other next-generation sequencing systems employ different measures according to their methods.

4.4.2 Repeats

In mammalian genomes the repetitive content can be as high as 50%. The repeated fraction includes interspersed repeats derived from transposable elements, and long genomic regions that have been duplicated in tandem, palindromic or dispersed fashion, *eg.* ribosomal RNA genes, centromeres, heterochromatin and retrotransposons. Such features complicate the assembly into a correct finished genome sequence and have a great influence on the design of assemblers. Computationally repeats are typically handled as follows:

- **Comparing:** By comparing reads to known repeated regions in other genomes, potential repetitive sequences can be separated (and typically discarded) from the assembly.
- **Masking:** Regions which have a high depth, that is regions where many reads share the same sequence, are marked as repeats (illustrated on Fig. 4). Usually such regions are discarded by the assembler, and are not incorporated in the assembly, *eg.* by the method presented in [137].

A standard program for masking repeats is RepeatMasker [138]. It searches through curated repeat databases (*eg.* Repbase [139]) using the alignment program cross_match [44] to identify and mask repeats. The speed of cross_match can be increased by using the software wrapper MaskerAid [140].

4.4.3 Expressed Sequence Tags

Due to the way that the EST sequences are generated, there are several concerns which can severely disrupt attempts to analyze the data:

Over-clustering: This happens when ESTs from different genes are clustered together, and therefore associated with the same genetic sequence. This often arise as a result of the cloning procedure, which falsely place two originally separate sequences in the same read, *ie.* chimerism. However, paralogous genes can also be clustered together due to high sequence similarity. Using the traditional (TGICL, d2_cluster) single transitive single linkage clustering methods [141, 142], can cause all EST from both genes to be assigned to the same cluster. More stringent clustering methods such as the double linkage of geneDistiller [54] can reduce the amount of falsely clustered reads, and create more consistent assemblies and consensus sequences [56].

Highly expressed genes: In non-normalized cDNA libraries the fraction of the genes that is highly expressed, will be represented in a high number and lead to large and deep clusters, that may accidentally contain EST from more than one gene. There are several ways to handle highly expressed genes depending on the purpose of the investigation: (i) Removal of known house keeping genes: If the sequence of some house keeping genes of the organism are known, removing ESTs that originate from these genes can alleviate the problems. (ii) Adding annotated gene sequences: If a genetic sequence of an annotated gene is known, it can be used as a template for the ESTs. (iii) Seeded clustering: Known full-length transcripts can be used for 'seeded clustering', which helps to create smaller, better partitioned clusters and avoid chimeric assemblies [130].

These procedures can alleviate some of the problems, however some clusters of highly expressed genes can still contain several thousands EST sequences. Producing a consensus sequence from such a large cluster can be tricky as most assembler are not able to handle such deep clusters. Several methods have been created to deal with this problem, such as the "containment clustering" of TGICL [130], or the alignment/consensus strategy of geneDistiller [54, 56].

Other minor concerns in EST assembly are overlapping genes *eg.* they can be on opposite strand and share a UTR-tail or have common motifs. This can cause the assembly program to assign ESTs from two different genes to the same cluster [52], and will complicate analysis of the cluster.

4.4.4 Reliability of high-throughput assemblies and sequence data

Although no major comparison of assemblies generated with different HTS technologies has been published yet, preliminary analysis shows that assemblies with 454 and Solexa significantly differ from those obtained with classical sequencing reads. In a survey of assemblies for *Streptcoccus suis* from 454, Solexa and capillary data, 454 sequencing of a library with 5-fold coverage produced 5336 contigs while the Sanger method, two-fold coverage, resulted in only 1011 contigs. The length of the largest contig was 5336 for 454 and 12257 for the capillary sequencing method. Moreover, using Solexa, a ten-fold coverage was necessary to produce 8370 contigs with a maximum length of only 1687 [143]. The best results were seen for hybrid assemblies comprising data from at least two different sequencing technologies. The authors concluded that assembly methods are to be refined to address the specific shortcomings of each method [143]. As mentioned earlier, the differences are likely to be caused by very different error patterns. In the case of the Solexa technology, error rates are highly position-dependent, variable across different machines and even across different runs [115]. In a recent investigation on the quality of Solexa reads, the authors found a bias in the read coverage: significantly more reads were found in GC-rich genomic intervals. Despite the manufactures specifications for the read quality, error rates varied from 0.3% to 3.8% [144]. Compared to 454 sequences, only few insertions and deletions were found [119]. In the future a new basecalling software, *eg.* Alta-Cyclic [145], might be able to improve the quality of Solexa sequences. Additionally, it has been shown that under idealized conditions it is theoretically possible to assemble bacterial genomes (with 80x coverage of 30 nt reads) [146].

4.5 Assembly of contigs - scaffolding

While the assembly of the individual reads into contigs give some (local) information, the contigs still need to be set into the context of the whole genome. This is carried in the last phase of an assembly process: scaffolding, which is the process

where different (genomic) contigs are organized into even larger frameworks (scaffolds or super-contigs). The contigs are ordered and oriented in a consistent way, so that the scaffold build is a true representation of chromosomes, though there may still be gaps between contigs, which are dealt with in new rounds of sequencing (see finishing below).

In the scaffolding stage of an assembly, all the information usually come from other sources than the reads themselves. This information includes read-pair information, STS (Sequence Tag Sites), and other sources [147].

4.6 Finishing

When an assembly has been completed, specific parts of the assembly usually need to be reexamined, perhaps due to low quality of the data, low (or no) coverage of the sequence, sites under suspicion of misassembly, etc. The reexamination are usually dealt with in an elaborate process where manual inspection is used to analyze the ambiguous section(s) and new ways are devised to clarify the particular ambiguities.

Analysis of the assembled contigs can be performed with a number of tools. One is Consed [148], which allows navigation of the assembled contigs and reads, problematic regions can be searched for with different criteria, and regions can be tagged for further inspection. Others are Autofinish [149], BACcardi [150], and GAP4 [151], all of which has different strengths and purposes.

4.7 Genome Resequencing

Recent developments in high-throughput sequencing technologies have ignited the scientific community's imagination. Terms such as the "personal genome" or "1000$ genome" are now popular in the media [152, 153]. The growing number of publicly available reference genomes allows genome resequencing on a larger scale, as sequencing costs decrease and throughput increases [154]. However, currently even HTS only allows deep resequencing of a small number of large individual genomes [155] or of specific parts of the genome. It has been remarked that the full power of high-throughput sequencers might not be unleashed since no suitable methods to select for specific genomic subsets are available and methods for targeted amplification are more likely to be effective [97]. However, recent methods using hybrid techniques such as microarray-based genomic selection (MGS) and multiplex exon capture to narrow down the number of sequences or to focus on specific genomic locations may overcome this shortcoming [98, 97]. Thanks to the contribution of James D. Watson a first complete personal genome, sequenced using 454, was published in 2008 [156]. In this project a set of 106.5 million reads, representing 24.5 billion bases and a depth of 7.4-fold, was generated. The mapped reads in combination with 454 quality values (Q-values) were used to gather a set of 3.32 million SNPs. Several filters had to be applied to increase specificity. A read was only included if the BLAT [65] alignment (i) was spanning at least 90% of the read length, (ii) did not have alternate hits, (iii) had less than five mismatches, (iv) had less than five indels. Subsequently, the remaining 93 million reads were again realigned with PHRAP's cross_match tool. Three additional filter steps using the quality score, (see supplementary material for [156]), the ratio of the variant to total coverage (> 0.2) and the vicinity to homopolymer runs ($< 5bp$) in order to avoid false positive indels ended this complicated procedure. Finally, the authors were able to discover approximately 500 000 new putative SNPs. Additionally, approximately 2.6 million reads of novel sequence and reads with low quality alignments were assembled in 170 000 contigs spanning 48Mb. After a filter step 110000 contigs spanning 29Mb remained [69]. The authors concluded that those contigs might represent the 25Mb predicted to be absent from the current reference genome. With costs of about 1 million US$, however, the "1000$ genome" genome still seems to be a distant prospect.

Next-generation HTS has also been applied for the mapping of translocation breakpoints. HTS not only reduces the labor and time cost of traditional methods in detecting translocation breakpoints, *eg.* in situ hybridization with fluorescent dye-labeled bacterial artificial chromosome clones (BAC-FISH), but also greatly improve the resolution so that the disrupted gene can be identified by PCR cloning. Thus, mapping and sequencing breakpoints region with Solexa platform has been used to identify novel candidate genes for mental retardation [93]. Probability calculations as well as simulations suggest that current paired-end sequencing technology already provides a high probability of breakpoint detection and good resolution in localizing structural chromosomal the rearrangements [125].

5 Overview of assembly methods

Different assemblers use different information in the assembly process. Some only use sequences in fasta format and the corresponding quality values, while others can assemble without quality values. Additional information on known sequences (*eg.* genes), clones, clone sizes and the orientation of the reads (forward-reverse) might be helpful in the assembly process.

An overview of different assemblers is presented in tables 1a and 1b, which summarizes the approach each program utilizes in assembly.

[Table 1a]
[Table 1b]

5.1 Assemblers

In the following a large selection of different assemblers that have been created over time are presented. An overview with shorts presentations of the different assemblers are given on the web-page `http://genome.ku.dk/resources/assembly/methods.html`.

One of the (relatively) early assemblers is PHRAP [44], which is still in use, both in itself (for small DNA sequence sets), and as a subcomponent of WGS assemblers, *eg.* RePS [157], Phusion [50], JAZZ [158], and ATLAS [159]. Other WGS assemblers that also use some variety of the standard overlap-layout-consensus approach are, the Celera assembler [45], CAP3 [116], RAMEN [160], PCAP [161], the TIGR assembler [162], STROLL [132], and ARACHNE2 [49]. Some new approaches to assembly have been attempted, among them mira [59] and TRAP [58], which try novel ways to deal with repetitive sequences by checking the trace and quality files. An emerging approach is to use more explicit graph based programs, such as Euler [133], Partial ordered alignment (POA) [163], Velvet [121], Splicing graphs [55], ASmodeler [164], and xtract [57], where the last three are used specifically for ESTs. Other programs that analyze ESTs are TGICL [165], StackPack [13], PaCE [166], Hidden Markov Model (HMM) Sampling [167], and geneDistiller [54]. Finally, some programs are used in the scaffolding stage, where contigs are processed and put in order, *eg.* GigAssembler [168] and Bambus [147] (part of the AMOS package [46]).

5.2 Assembler Comparisons

Comparing the different assemblers is not a trivial task due to several factors. Not to mention the problems of constructing appropriate benchmark data. First the different assemblers use a variety of input data, and so comparing an assembler which uses a lot of the additional information to one which only uses a fragment of the information is inappropriate. Another aspect is evaluating the success criteria, the goal is to create a single error-free contig of each chromosome, which means that fewer gaps, longer contigs, and fewer errors are desired. However, different assemblers might do better in one area and worse in another, so weighing the performance of one assembler against another can be difficult. Still there have been a few attempts to compare assemblers.

In [132], PHRAP, TIGR Assembler, and STROLL were compared on sequence data from the bacterium *Borrelia burgdorferi*. Phusion and ARACHNE were both applied to the assembly of the Mouse genome [169, 50]. PHRAP has been compared to CAP3 in [116] (on four BAC datasets) and [76] (on EST data) where the TIGR Assembler was also included. Furthermore, a short comparison between PHRAP, Arachne, and Euler is presented in [60].

Common to these studies is that the individual performance of the assemblers depend on the data they are presented with. PHRAP is generally aggressive in joining reads and creates large contigs, though sometimes at the expense of introducing errors. This assembler would be a fairly good choice if the dataset consisted only of reads with assigned quality values. However if additional information, such as forward-reverse constraints, is available other programs (*eg.* CAP3, STROLL) would perform better. Another observation is that the performance of PHRAP degrades when it is applied to some large data sets. Additionally an updated assembler based on the Euler package [60], Euler-SR [115], is available. Euler-SR

which uses a revised version of the Euler package, is less space intensive and optimized for short Reads. Alternatives are assemblers such as Arachne or Velvet [121].

6 Applying assemblies for other analyzes

There are different possibilities for further processing of the data and thereby for finding interesting and important features for future investigation, for example searching for SNPs (Single Nucleotide Polymorphisms) and alternative splice forms, or comparing genomes with each other.

SNP detection: ESTs are the most often used data source for SNP detection, but SNPs can be found from shotgun data as well. SNPs in transcribed sequences can either be synonymous (no amino acid change), or non-synonymous (encoding a different amino acid). A variety of different computer programs are designed for SNPs analysis. Some find and predict whether a given site is polymorphic, *eg.* Polybayes [170], Polyphred [171] and novoSNP [172]. Others try to predict whether a given SNP is potentially harmful or neutral, *eg.* Polyphen [173] and SIFT [174].

Massively parallel Sequencing The new massively parallel sequencing technologies will provide a wealth of new information. As mentioned above they have already been applied for the sequencing of an individuals genome [156], and detection of genomic rearrangements [93, 125], and in the future new ways of utilizing their enormous capacity will likely appear, both with respect to the number of clones that are analyzed and the total amount of sequenced DNA.

Detection of alternative splicing: In eukaryotes, the removal of introns by splicing is a crucial step in gene expression. For some genes, splicing results in only one single type of mRNA, but studies have revealed that up to 60% of the human genes result in two or more mRNA isoforms due to alternative splicing [36, 175]. One approach to investigate alternative splicing is through assemblies of ESTs. However, assemblies of ESTs usually has multiple solutions in the presence of alternative splicing, which might end in truncated, misassembled or missing transcripts [175, 176]. Having a completed genome as a reference can help because it allows comparison of the EST to the corresponding genomic sequence. Some programs have been created which explicitly try to address the problem of assembling alternative splice variants from ESTs, among them are Splicing Graph [55] and geneDistiller [54].

Genome Comparison: Furthermore, as different sequencing project complete their respective genomes and the data become available, it becomes possible to compare differences and similarities between different species on a sequence basis. This can generate a wealth of new information, and give new insights into the evolution and biology of living organisms. Examples of how such a comparative analysis can be performed are given in [177, 67, 62].

7 Discussion

As still more genomes are studied and more sophisticated computer programs for genome assembly and analysis are developed, our knowledge of genomics will expand tremendously. Sequencing technologies have already given us a consensus sequence of *homo sapiens*, and in the future we can expect that many individual human genomes will be sequenced , which will add to the steadily growing number of genetic variations and genetic predisposition to disease that has been revealed in our specie. Furthermore, many model organisms and eventually, all species remain to be sequenced, which will give a better understanding of life and its evolution.

For mammalian genomes whole genome shotgun sequencing is likely to entail similar costs for producing a finished sequence as a hierarchical shotgun solution. The hierarchical approach has a higher initial cost than the whole-genome approach, owing to the need to create a map of clones (about 1% of the total cost of sequencing) and to identify sequence overlap between clones. On the other hand, the whole-genome approach is likely to require much greater work and expense in the final stage of the assembly, because of the challenge of resolving misassemblies.

New high-throughput sequencing technologies have rapidly emerged. However, the sequencing methods as well as the computational tools have to be further improved, to allow a complete de novo assembly for large genomes with these technologies. However, today only little data on the error models of different massively parallel sequencing technologies is available. These error models are crucial to interpret and analyze the sequence data correctly [144]. When it comes to

de novo assembly, the short read lengths of SOLiD and Solexa methodologies seem to be a momentous disadvantage and the high number of reads produced might not be able to compensate for this handicap. However, all manufacturers aim to increase the read lengths. Currently, a reasonable approach to the assembly of such short sequences could include data from low coverage Sanger sequencing. Although hybrid data set approaches are cumbersome [115], they have already been shown to produce useful assemblies [124].

The choice of sequencing strategy should also be influenced by the goal of the project. In some organisms it might be desirable to quickly generate a few contigs covering key points in the genome, while in others a broader strategy might apply. Still other projects combine whole–genome with hierarchical shotgun in a hybrid approach trying to utilize the strengths of each [159].

Other applications of sequencing and assembly are continuously being explored. For example, the growing field of environmental sequencing (or metagenomics) [178, 179, 180], will undoubtedly present new challenges to assemblers, since sequence data will no longer be known to come from a single source organism, but from several and often from a multitude of distinct organisms, with different relative abundances, different genome structures, repeat content, and so on. A somewhat related field is paleogenomics – is sequencing of fossil DNA. This field has become much more accessible with the new massively parallel sequencing methods, as the traditional Sanger sequencing is difficult and technical impractical on fossil DNA samples. The new techniques, however, have made it possible to extract genomic information from long extinct species, for example the woolly mammoth [181].

The assemblers presented in this paper show the great diversity and ingenuity that has gone into finding better ways of assembling the DNA puzzle from diverse types of data. The various strategies for overcoming the challenges revealed in assembly are also discussed. Newer assemblers (and associated programs) endeavor to surmount these challenges in novel ways, and it is likely that computational whole genome assembly will be further refined in the future. Also, it should be remembered, that a substantial fraction of the large genomes still evades sequencing/assembly with existing technology [69]. The estimated $\sim$10% of the human genome which has not been sequenced may not be without function, as exemplified by the centromeres and pericentric heterochromatic regions. Many of the tandem repeats within these regions have been sequenced at clone scale, but none have been sequenced at genome-scale, where their size exceeding many megabases preclude assembly. Why the remaining >250 smaller gaps, scattered over the euchromatic part of the human genome, with sizes ranging from 20 to 100 kb, cannot be sequenced/assembled is unknown. It is likely that this terra incognita will only be sequenced when (if) single molecule, very long read sequencing technologies have been developed.

Acknowledgments

Thanks to Anders Blomberg for comments on a early version of the manuscript. KSA was supported by a grant from the Faculty of Life Sciences, University of Copenhagen. This work was further supported by the Danish research council FTP and the Danish Center for Scientific Computing. The Wilhelm Johannsen Centre for Functional Genome Research is established and funded by the Danish National Research Foundation.

References

[1] K. Liolios, K. Mavromatis, N. Tavernarakis, N. Kyrpides, The genomes on line database (gold) in 2007: status of genomic and metagenomic projects and their associated metadata., Nucleic Acids Res 36 (Database Issue) (2008) D475–9.

[2] F. Sanger, G. Air, B. Barrell, N. Brown, A. Coulson, C. Fiddes, C. Hutchison, P. Slocombe, M. Smith, Nucliotide sequence of bacteriophage phi X174 DNA., Nature 265 (5596) (1977) 687–95.

[3] F. Sanger, A. Coulson, T. Friedmann, G. Air, B. Barrell, N. Brown, J. Fiddes, C. r. Hutchison, P. Slocombe, M. Smith, The nucleotide sequence of bacteriophage phiX174., J Mol Biol 125 (2) (1978) 225–46.

[4] F. Sanger, A. Coulson, G. Hong, D. Hill, G. Petersen, Nucleotide sequence of bacteriophage lambda DNA., J Mol Biol 162 (4) (1982) 729–73.

[5] W. Fiers, R. Contreras, G. Haegemann, R. Rogiers, A. Van de Voorde, H. Van Heuverswyn, J. Van Herreweghe, G. Volckaert, M. Ysebaert, Complete nucleotide sequence of SV40 DNA., Nature 273 (5658) (1978) 113–20.

[6] S. Anderson, A. Bankier, B. Barrell, M. de Bruijn, A. Coulson, J. Drouin, I. Eperon, D. Nierlich, B. Roe, F. Sanger, et al., Sequence and organization of the human mitochondrial genome., Nature 290 (5806) (1981) 457–65.

[7] S. Anderson, Shotgun DNA sequencing using cloned DNase I-generated fragments., Nucleic Acids Res 9 (13) (1981) 3015–27.

[8] P. Deininger, Random subcloning of sonicated DNA: application to shotgun DNA sequence analysis., Anal Biochem 129 (1) (1983) 216–23.

[9] A. Edwards, H. Voss, P. Rice, A. Civitello, J. Stegemann, C. Schwager, J. Zimmermann, H. Erfle, C. Caskey, W. Ansorge, Automated DNA sequencing of the human HPRT locus., Genomics 6 (4) (1990) 593–608.

[10] R. Wooster, Identification of the breast cancer susceptibility gene BRCA2., Nature 378 (1995) 789–92.

[11] R. Fleischmann, M. Adams, O. White, R. Clayton, E. Kirkness, A. Kerlavage, C. Bult, J. Tomb, B. Dougherty, J. Merrick, et al.;, Whole-genome random sequencing and assembly of Haemophilus influenzae Rd., Science 269 (5223) (1995) 496–512.

[12] M. Adams, J. Kelley, J. Gocayne, M. Dubnick, M. Polymeropoulos, H. Xiao, C. Merril, A. Wu, B. Olde, R. Moreno, et al, Complementary DNA sequencing: expressed sequence tags and human genome project., Science. 252 (5013) (1991) 1651–6.

[13] A. Christoffels, A. van Gelder, G. Greyling, R. Miller, T. Hide, W. Hide, STACK: Sequence Tag Alignment and Consensus Knowledgebase., Nucleic Acids Res 29 (1) (2001) 234–8.

[14] M. Boguski, The turning point in genome research., Trends Biochem Sci. 20 (8) (1995) 295–6.

[15] M. Marra, L. Hillier, R. Waterston, Expressed sequence tags–ESTablishing bridges between genomes., Trends Genet. 14 (1) (1998) 4–7.

[16] M. Adams, M. Dubnick, A. Kerlavage, R. Moreno, J. Kelley, T. Utterback, J. Nagle, C. Fields, J. Venter, Sequence identification of 2,375 human brain genes., Nature. 355 (6361) (1992) 632–4.

[17] M. Adams, A. Kerlavage, C. Fields, J. Venter, 3,400 new expressed sequence tags identify diversity of transcripts in human brain., Nat Genet. 4 (3) (1993) 256–67.

[18] T. Nakamura, G. Morin, K. Chapman, S. Weinrich, W. Andrews, J. Lingner, C. Harley, T. Cech, Telomerase catalytic subunit homologs from fission yeast and human., Science. 277 (5328) (1997) 955–9.

[19] R. Medzhitov, P. Preston Hurlburt, C. J. Janeway, A human homologue of the Drosophila Toll protein signals activation of adaptive immunity., Nature. 388 (6640) (1997) 394–7.

[20] F. Liang, I. Holt, G. Pertea, S. Karamycheva, S. Salzberg, J. Quackenbush, Gene index analysis of the human genome estimates approximately 120,000 genes., Nat Genet. 25 (2) (2000) 239–40.

[21] T. Hudson, L. Stein, S. Gerety, J. Ma, A. Castle, J. Silva, D. Slonim, R. Baptista, L. Kruglyak, S. Xu, et al.;, An STS-based map of the human genome., Science. 270 (5244) (1995) 1945–54.

[22] G. Schuler, M. Boguski, E. Stewart, L. Stein, G. Gyapay, K. Rice, R. White, P. Rodriguez Tome, A. Aggarwal, E. Bajorek, et al., A gene map of the human genome., Science. 274 (5287) (1996) 540–6.

[23] P. Deloukas, G. Schuler, G. Gyapay, E. Beasley, C. Soderlund, P. Rodriguez Tome, L. Hui, T. Matise, K. McKusick, Beckmann, et al., A physical map of 30,000 human genes., Science. 282 (5389) (1998) 744–6.

[24] R. Waterston, C. Martin, M. Craxton, C. Huynh, A. Coulson, L. Hillier, R. Durbin, P. Green, R. Shownkeen, N. Halloran, et al.;, A survey of expressed genes in Caenorhabditis elegans., Nat Genet. 1 (2) (1992) 114–23.

[25] W. McCombie, M. Adams, J. Kelley, M. FitzGerald, T. Utterback, M. Khan, M. Dubnick, A. Kerlavage, J. Venter, C. Fields, Caenorhabditis elegans expressed sequence tags identify gene families and potential disease gene homologues., Nat Genet. 1 (2) (1992) 124–31.

[26] L. Brody, K. Abel, L. Castilla, F. Couch, D. McKinley, G. Yin, P. Ho, S. Merajver, S. Chandrasekharappa, J. Xu, et al.;, Construction of a transcription map surrounding the BRCA1 locus of human chromosome 17., Genomics. 25 (1) (1995) 238–47.

[27] Z. Kan, E. Rouchka, W. Gish, D. States, Gene structure prediction and alternative splicing analysis using genomically aligned ESTs., Genome Res. 11 (5) (2001) 889–900.

[28] S. Tugendreich, D. J. Bassett, V. McKusick, M. Boguski, P. Hieter, Genes conserved in yeast and humans., Hum Mol Genet. 3 Spec No (1994) 1509–17.

[29] N. Papadopoulos, N. Nicolaides, Y. Wei, S. Ruben, K. Carter, C. Rosen, W. Haseltine, R. Fleischmann, C. Fraser, M. Adams, et al.;, Mutation of a mutL homolog in hereditary colon cancer., Science. 263 (5153) (1994) 1625–9.

[30] M. Adams, A. Kerlavage, R. Fleischmann, R. Fuldner, C. Bult, N. Lee, E. Kirkness, K. Weinstock, J. Gocayne, O. White, et al.;, Initial assessment of human gene diversity and expression patterns based upon 83 million nucleotides of cDNA sequence., Nature (1995) 3–174.

[31] R. Braren, K. Firner, S. Balasubramanian, F. Bazan, H. Thiele, F. Haag, F. Koch Nolte, Use of the EST database resource to identify and clone novel mono(ADP-ribosyl)transferase gene family members., Adv Exp Med Biol. 419 (1997) 163–8.

[32] R. Allikmets, B. Gerrard, D. Glavac, M. Ravnik Glavac, N. Jenkins, D. Gilbert, N. Copeland, W. Modi, M. Dean, Characterization and mapping of three new mammalian ATP-binding transporter genes from an EST database., Mamm Genome. 6 (2) (1995) 114–7.

[33] P. Nelson, D. Han, Y. Rochon, G. Corthals, B. Lin, A. Monson, V. Nguyen, B. Franza, S. Plymate, R. Aebersold, et al., Comprehensive analyses of prostate gene expression: convergence of expressed sequence tag databases, transcript profiling and proteomics., Electrophoresis. 21 (9) (2000) 1823–31.

[34] K. Garg, P. Green, D. Nickerson, Identification of candidate coding region single nucleotide polymorphisms in 165 human genes using assembled expressed sequence tags., Genome Res. 9 (11) (1999) 1087–92.

[35] K. Buetow, M. Edmonson, A. Cassidy, Reliable identification of large numbers of candidate SNPs from public EST data., Nat Genet. 21 (3) (1999) 323–5.

[36] D. Brett, J. Hanke, G. Lehmann, S. Haase, S. Delbruck, S. Krueger, J. Reich, P. Bork, EST comparison indicates 38contain possible alternative splice forms., FEBS Lett. 474 (1) (2000) 83–6.

[37] A. Mironov, J. Fickett, M. Gelfand, Frequent alternative splicing of human genes., Genome Res. 9 (12) (1999) 1288–93.

[38] R. Sorek, H. Safer, A novel algorithm for computational identification of contaminated EST libraries., Nucleic Acids Res. 31 (3) (2003) 1067–74.

[39] . International Human Genome Sequencing Consortium, Initial sequencing and analysis of the human genome., Nature. 409 (6822) (2001) 860–921.

[40] J. Venter, M. Adams, E. Myers, P. Li, R. Mural, G. Sutton, H. Smith, M. Yandell, C. Evans, R. Holt, et al., The sequence of the human genome., Science. 291 (5507) (2001) 1304–51.

[41] R. Staden, A new computer method for the storage and manipulation of DNA gel reading data., Nucleic Acids Res 8 (16) (1980) 3673–94.

[42] H. Peltola, H. Soderlund, E. Ukkonen, SEQAID: a DNA sequence assembling program based on a mathematical model., Nucleic Acids Res. 12 (1984) 307–21.

[43] X. Huang, A contig assembly program based on sensitive detection of fragment overlaps., Genomics. 14 (1) (1992) 18–25.

[44] Green Laboratory, Phred, phrap, consed documentation, http://www.phrap.org/phredphrapconsed.html (1994).

[45] E. Myers, G. Sutton, A. Delcher, I. Dew, D. Fasulo, M. Flanigan, S. Kravitz, C. Mobarry, K. Reinert, K. Remington, et al., A whole-genome assembly of Drosophila., Science. 287 (5461) (2000) 2196–204.

[46] AMOS consortium, Amos open-source assembler, http://amos.sourceforge.net/ (–).

[47] S. Aparicio, J. Chapman, E. Stupka, N. Putnam, J. Chia, P. Dehal, A. Christoffels, S. Rash, S. Hoon, A. Smit, et al., Whole-genome shotgun assembly and analysis of the genome of Fugu rubripes., Science 297 (5585) (2002) 1301–10.

[48] P. Dehal, Y. Satou, R. Campbell, J. Chapman, B. Degnan, A. De Tomaso, B. Davidson, A. Di Gregorio, M. Gelpke, D. Goodstein, et al, The draft genome of Ciona intestinalis: insights into chordate and vertebrate origins., Science. 298 (5601) (2002) 2157–67.

[49] S. Batzoglou, D. Jaffe, K. Stanley, J. Butler, S. Gnerre, E. Mauceli, B. Berger, J. Mesirov, E. Lander, ARACHNE: a whole-genome shotgun assembler., Genome Res 12 (1) (2002) 177–89.

[50] J. Mullikin, Z. Ning, The phusion assembler., Genome Res. 13 (1) (2003) 81–90.

[51] R. Miller, A. Christoffels, C. Gopalakrishnan, J. Burke, A. Ptitsyn, T. Broveak, W. Hide, A comprehensive approach to clustering of expressed human gene sequence: the sequence tag alignment and consensus knowledge base., Genome Res 9 (11) (1999) 1143–55.

[52] J. Burke, H. Wang, W. Hide, D. Davison, Alternative gene form discovery and candidate gene selection from gene indexing projects., Genome Res 8 (3) (1998) 276–90.

[53] J. Quackenbush, J. Cho, D. Lee, F. Liang, I. Holt, S. Karamycheva, B. Parvizi, G. Pertea, R. Sultana, J. White, The TIGR Gene Indices: analysis of gene transcript sequences in highly sampled eukaryotic species., Nucleic Acids Res 29 (1) (2001) 159–64.

[54] M. Gilchrist, A. Zorn, J. Voigt, J. Smith, N. Papalopulu, E. Amaya, Defining a large set of full-length clones from a Xenopus tropicalis EST project., Dev Biol. 271 (2) (2004) 498–516.

[55] S. Heber, M. Alekseyev, S. Sze, H. Tang, P. Pevzner, Splicing graphs and EST assembly problem., Bioinformatics 18 Suppl 1 (2002) S181–8.

[56] K. Scheibye-Alsing, M. Gilchrist, J. Gorodkin, EST assembly with genedistiller, *In preparation*.

[57] K. Malde, E. Coward, I. Jonassen, A graph based algorithm for generating EST consensus sequences., Bioinformatics 21 (8) (2005) 1371–5.

[58] M. Tammi, E. Arner, B. Andersson, TRAP: Tandem Repeat Assembly Program produces improved shotgun assemblies of repetitive sequences., Comput Methods Programs Biomed. 70 (1) (2003) 47–59.

[59] B. Chevreux, T. Pfisterer, B. Drescher, A. Driesel, W. Muller, T. Wetter, S. Suhai, Using the miraEST assembler for reliable and automated mRNA transcript assembly and SNP detection in sequenced ESTs., Genome Res. 14 (6) (2004) 1147–59.

[60] P. Pevzner, H. Tang, G. Tesler, De novo repeat classification and fragment assembly., Genome Res. 14 (9) (2004) 1786–96.

[61] A. Delcher, A. Phillippy, J. Carlton, S. Salzberg, Fast algorithms for large-scale genome alignment and comparison., Nucleic Acids Res 30 (11) (2002) 2478–83.

[62] K. Makino, K. Oshima, K. Kurokawa, K. Yokoyama, T. Uda, K. Tagomori, Y. Iijima, M. Najima, M. Nakano, A. Yamashita, et al, Genome sequence of Vibrio parahaemolyticus: a pathogenic mechanism distinct from that of V cholerae., Lancet 361 (9359) (2003) 743–9.

[63] P. Flicek, E. Keibler, P. Hu, I. Korf, M. Brent, Leveraging the mouse genome for gene prediction in human: from whole-genome shotgun reads to a global synteny map., Genome Res 13 (1) (2003) 46–54.

[64] A. Tenney, R. Brown, C. Vaske, J. Lodge, T. Doering, M. Brent, prediction and verification in a compact genome with numerous small introns., Genome Res. 14 (11) (2004) 2330–5.

[65] W. Kent, BLAT–the BLAST-like alignment tool., Genome Res 12 (4) (2002) 656–64.

[66] N. Bray, I. Dubchak, L. Pachter, AVID: A global alignment program., Genome Res 13 (1) (2003) 97–102.

[67] O. Couronne, A. Poliakov, N. Bray, T. Ishkhanov, D. Ryaboy, E. Rubin, L. Pachter, I. Dubchak, Strategies and tools for whole-genome alignments., Genome Res 13 (1) (2003) 73–80.

[68] A. Lefebvre, T. Lecroq, H. Dauchel, J. Alexandre, FORRepeats: detects repeats on entire chromosomes and between genomes., Bioinformatics 19 (3) (2003) 319–26.

[69] International Human Genome Sequencing Consortium., Finishing the euchromatic sequence of the human genome., Nature 431 (7011) (2004) 931–45.

[70] F. Sanger, S. Nicklen, A. R. Coulson, Dna sequencing with chain-terminating inhibitors, Proc Natl Acad Sci U S A 74 (12) (1977) 5463–5467.

[71] L. Rowen, G. Mahairas, L. Hood, Sequencing the human genome., Science 278 (5338) (1997) 605–7.

[72] G. Churchill, M. Waterman, The accuracy of DNA sequences: estimating sequence quality., Genomics. 14 (1) (1992) 89–98.

[73] M. Giddings, R. J. Brumley, M. Haker, L. Smith, An adaptive, object oriented strategy for base calling in DNA sequence analysis., Nucleic Acids Res. 21 (19) (1993) 4530–40.

[74] C. Lawrence, V. Solovyev, Assignment of position-specific error probability to primary DNA sequence data., Nucleic Acids Res. 22 (7) (1994) 1272–80.

[75] R. Lipshutz, F. Taverner, K. Hennessy, G. Hartzell, R. Davis, DNA sequence confidence estimation., Genomics. 19 (3) (1994) 417–24.

[76] F. Liang, I. Holt, G. Pertea, S. Karamycheva, S. Salzberg, J. Quackenbush, An optimized protocol for analysis of EST sequences., Nucleic Acids Res 28 (18) (2000) 3657–65.

[77] F. Sanger, A. Coulson, A rapid method for determining sequences in DNA by primed synthesis with DNA polymerase., J Mol Biol 94 (3) (1975) 441–8.

[78] E. Chen, D. Schlessinger, J. Kere, Ordered shotgun sequencing, a strategy for integrated mapping and sequencing of YAC clones., Genomics. 17 (3) (1993) 651–6.

[79] M. Smith, A. Holmsen, Y. Wei, M. Peterson, G. Evans, Genomic sequence sampling: a strategy for high resolution sequence-based physical mapping of complex genomes., Nat Genet. 7 (1) (1994) 40–7.

[80] K. Kupfer, M. Smith, J. Quackenbush, G. Evans, Physical mapping of complex genomes by sampled sequencing: a theoretical analysis., Genomics. 27 (1) (1995) 90–100.

[81] J. Roach, C. Boysen, K. Wang, L. Hood, Pairwise end sequencing: a unified approach to genomic mapping and sequencing., Genomics. 26 (2) (1995) 345–53.

[82] D. Nurminsky, D. Hartl, Sequence scanning: A method for rapid sequence acquisition from large-fragment DNA clones., Proc Natl Acad Sci U S A. 93 (4) (1996) 1694–8.

[83] J. Weber, E. Myers, Human whole-genome shotgun sequencing., Genome Res. 7 (5) (1997) 401–9.

[84] M. Marra, T. Kucaba, N. Dietrich, E. Green, B. Brownstein, R. Wilson, K. McDonald, L. Hillier, J. McPherson, R. Waterston, High throughput fingerprint analysis of large-insert clones., Genome Res. 7 (11) (1997) 1072–84.

[85] Rat Genome Sequencing Project Consortium, Genome sequence of the Brown Norway rat yields insights into mammalian evolution., Nature. 428 (6982) (2004) 493–521.

[86] D. Altshuler, V. Pollara, C. Cowles, W. Van Etten, J. Baldwin, L. Linton, E. Lander, An SNP map of the human genome generated by reduced representation shotgun sequencing, Nature 407 (2000) 513–516.

[87] N. Springer, X. Xu, W. Barbazuk, Utility of different gene enrichment approaches toward identifying and sequencing the maize gene space, Plant Physiology Preview 136 (2004) 3023–3033.

[88] J. Bedell, M. Budiman, A. Nunberg, R. Citek, D. Robbins, J. Jones, E. Flick, T. Rholfing, J. Fries, K. Bradford, J. McMenamy, M. Smith, H. Holeman, B. Roe, G. Wiley, I. Korf, P. Rabinowicz, N. Lakey, W. McCombie, J. Jeddeloh, R. Martienssen, Sorghum genome sequencing by methylation filtration, PLoS Biology 3 (2005) e13.

[89] W. Barbazuk, J. Bedell, P. Rabinowicz, Reduced representation sequencing: a success in maize and a promise for other plant genomes., Bioessays. 27 (8) (2005) 839–48.

[90] E. Glazov, P. Cottee, W. Barris, R. Moore, B. Dalrymple, M. Tizard, A microRNA catalog of the developing chicken embryo identified by a deep sequencing approach, Genome Res 18 (6) (2008) 957–64.

[91] R. Morin, M. O'Connor, M. Griffith, F. Kuchenbauer, A. Delaney, A. Prabhu, Y. Zhao, H. McDonald, T. Zeng, M. Hirst, C. Eaves, M. Marra, Application of massively parallel sequencing to microRNA profiling and discovery in human embryonic stem cells, Genome Res 18 (4) (2008) 610–21.

[92] A. Meissner, T. Mikkelsen, H. Gu, M. Wernig, J. Hanna, A. Sivachenko, X. Zhang, B. Bernstein, C. Nusbaum, D. Jaffe, A. Gnirke, R. Jaenisch, E. Lander, Genome-scale dna methylation maps of pluripotent and differentiated cells, Nature 454 (7205) (2008) 766–70.

[93] W. Chen, V. Kalscheuer, A. Tzschach, C. Menzel, R. Ullmann, M. Schulz, F. Erdogan, N. Li, Z. Kijas, G. Arkesteijn, I. Pajares, M. Goetz-Sothmann, U. Heinrich, I. Rost, A. Dufke, U. Grasshoff, B. Glaeser, M. Vingron, H. Ropers, Mapping translocation breakpoints by next-generation sequencing, Genome Res 18 (7) (2008) 1143–9.

[94] D. Schones, K. Zhao, Genome-wide approaches to studying chromatin modifications, Nat Rev Genet 9 (3) (2008) 179–91.

[95] C. Schmid, P. Bucher, Chip-seq data reveal nucleosome architecture of human promoters, Cell 131 (5) (2007) 831–2.

[96] M. ER, ChIP-seq: welcome to the new frontier, Nat Methods 4 (8) (2007) 613–4.

[97] G. Porreca, K. Zhang, J. Li, B. Xie, D. Austin, S. Vassallo, E. Leproust, B. Peck, C. Emig, F. Dahl, Y. Gao, G. Church, J. Shendure, Multiplex amplification of large sets of human exons, Nature Methods 4 (11) (2007) 931–936. doi:http://dx.doi.org/10.1038/nmeth1110.
URL http://dx.doi.org/10.1038/nmeth1110

[98] D. Okou, K. Steinberg, C. Middle, D. Cutler, T. Albert, M. Zwick, Microarray-based genomic selection for high-throughput resequencing., Nat Methodsdoi:http://dx.doi.org/10.1038/nmeth1109.
URL http://dx.doi.org/10.1038/nmeth1109

[99] C. Van Tassell, T. Smith, L. Matukumalli, J. Taylor, R. Schnabel, C. Lawley, C. Haudenschild, S. Moore, W. Warren, T. Sonstegard, Snp discovery and allele frequency estimation by deep sequencing of reduced representation libraries, Nat Methods 5 (3) (2008) 247–52.

[100] E. Hodges, Z. Xuan, V. Balija, M. Kramer, M. Molla, S. Smith, C. Middle, M. Rodesch, T. Albert, G. Hannon, W. McCombie, Genome-wide in situ exon capture for selective resequencing, Nat Genet 39 (12) (2007) 1522–7.

[101] G. Schuler, Pieces of the puzzle: expressed sequence tags and the catalog of human genes., J Mol Med. 75 (10) (1997) 694–8.

[102] S. Nagaraj, R. Gasser, S. Ranganathan, A hitchhiker's guide to expressed sequence tag (est) analysis, Brief Bioinform 8 (1) (2007) 6–21.

[103] M. Ronaghi, M. Uhlen, P. Nyren, A Sequencing Method Based on Real-Time Pyrophosphate., Science 281 (5375) (1998) 363–5.

[104] M. Margulies, M. Egholm, W. Altman, S. Attiya, J. Bader, L. Bemben, J. Berka, M. Braverman, Y. Chen, Z. Chen, S. Dewell, L. Du, J. Fierro, X. Gomes, B. Godwin, W. He, S. Helgesen, C. Ho, G. Irzyk, S. Jando, M. Alenquer, T. Jarvie, K. Jirage, J. Kim, J. Knight, J. Lanza, J. Leamon, S. Lefkowitz, M. Lei, J. Li, K. Lohman, H. Lu, V. Makhijani, K. McDade, M. McKenna, E. Myers, E. Nickerson, J. Nobile, R. Plant, B. Puc, M. Ronan, G. Roth, G. Sarkis, J. Simons, J. Simpson, M. Srinivasan, K. Tartaro, A. Tomasz, K. Vogt, G. Volkmer, S. Wang, Y. Wang, M. Weiner, P. Yu, R. Begley, J. Rothberg, Genome sequencing in microfabricated high-density picolitre reactors., Nature 437 (7057) (2005) 376–380. doi:10.1038/nature03959.
URL http://dx.doi.org/10.1038/nature03959

[105] S. Bennett, Solexa ltd, Pharmacogenomics 5 (4) (2004) 433–438. doi:10.1517/14622416.5.4.433.
URL http://www.futuremedicine.com/doi/abs/10.1517/14622416.%5.4.433

[106] A. Valouev, J. Ichikawa, T. Tonthat, J. Stuart, S. Ranade, H. Peckham, K. Zeng, J. Malek, G. Costa, K. McKernan, A. Sidow, A. Fire, S. Johnson, A high-resolution, nucleosome position map of C. elegans reveals a lack of universal sequence-dictated positioning, Genome Res. 18 (7) (2008) 1051–1063. arXiv:http://genome.cshlp.org/cgi/reprint/18/7/1051.pdf, doi:10.1101/gr.076463.108.
URL http://genome.cshlp.org/cgi/content/abstract/18/7/1051

[107] T. Harris, P. Buzby, H. Babcock, E. Beer, J. Bowers, I. Braslavsky, M. Causey, J. Colonell, J. Dimeo, J. Efcavitch, E. Giladi, J. Gill, J. Healy, M. Jarosz, D. Lapen, K. Moulton, S. Quake, K. Steinmann, E. Thayer, A. Tyurina, R. Ward, H. Weiss, Z. Xie, Single-Molecule DNA Sequencing of a Viral Genome, Science 320 (5872) (2008) 106–109. arXiv:http://www.sciencemag.org/cgi/reprint/320/5872/106.pdf, doi:10.1126/science.1150427.
URL http://www.sciencemag.org/cgi/content/abstract/320/5872%/106

[108] D. Fologea, M. Gershow, B. Ledden, D. McNabb, J. Golovchenko, J. Li, Detecting single stranded dna with a solid state nanopore, Nano Letters 5 (10) (2005) 1905–1909.
URL http://pubs3.acs.org/acs/journals/doilookup?in_doi=10.1%021/nl051199m

[109] C. Dekker, Solid-state nanopores., Nat Nanotechnol 2 (4) (2007) 209–215. doi:10.1038/nnano.2007.27.
URL http://dx.doi.org/10.1038/nnano.2007.27

[110] H. Li, J. Ruan, R. Durbin, Mapping short DNA sequencing reads and calling variants using mapping quality scores, Genome Res. (2008) gr.078212.108arXiv:http://genome.cshlp.org/cgi/reprint/gr.078212.108v1.pdf, doi:10.1101/gr.078212.108.
URL http://genome.cshlp.org/cgi/content/abstract/gr.078212.%108v1

[111] R. Li, Y. Li, K. Kristiansen, J. Wang, SOAP: short oligonucleotide alignment program, Bioinformatics (2008) btn025doi:10.1093/bioinformatics/btn025.
URL http://bioinformatics.oxfordjournals.org/cgi/content/ab%stract/btn025v1

[112] S. Rumble, M. Brudno, P. Lacroute, V. Yanovsky, M. Fiume, A. Dalca, Shrimp, http://compbio.cs.toronto.edu/shrimp.

[113] M. Pop, Shotgun sequence assembly, Advances in computers 60 (2004) 193–248.

[114] A. Sundquist, M. Ronaghi, H. Tang, P. Pevzner, S. Batzoglou, , PLoS ONE 2 (5) (2007) e484. doi:10.1371/journal.pone.0000484.
URL http://dx.doi.org/10.1371%2Fjournal.pone.0000484

[115] M. Chaisson, P. Pevzner, Short read fragment assembly of bacterial genomes, Genome Res. 18 (2) (2008) 324–330. arXiv:http://genome.cshlp.org/cgi/reprint/18/2/324.pdf, doi:10.1101/gr.7088808.
URL http://genome.cshlp.org/cgi/content/abstract/18/2/324

[116] X. Huang, A. Madan, CAP3: A DNA sequence assembly program., Genome Res 9 (9) (1999) 868–77.

[117] B. Ewing, P. Green, Base-calling of automated sequencer traces using phred. II. Error probabilities., Genome Res 8 (3) (1998) 186–94.

[118] Daniel H. Wagner Associates, Cats basecaller, http://www.wagner.com/technologies/biotech/catsadcopy.html (–).

[119] S. Huse, J. Huber, H. Morrison, M. Sogin, D. Welch, Accuracy and quality of massively parallel dna pyrosequencing, Genome Biology 8 (7) (2007) R143. doi:10.1186/gb-2007-8-7-r143.
URL http://genomebiology.com/2007/8/7/R143

[120] R. Cronn, A. Liston, M. Parks, D. Gernandt, R. Shen, T. Mockler, Multiplex sequencing of plant chloroplast genomes using Solexa sequencing-by-synthesis technology, Nucl. Acids Res. (2008) gkn502arXiv:http://nar.oxfordjournals.org/cgi/reprint/gkn502v1.pdf, doi:10.1093/nar/gkn502.
URL http://nar.oxfordjournals.org/cgi/content/abstract/gkn5%02v1

[121] D. Zerbino, E. Birney, Velvet: Algorithms for de novo short read assembly using de Bruijn graphs, Genome Res. 18 (5) (2008) 821–829. arXiv:http://genome.cshlp.org/cgi/reprint/18/5/821.pdf, doi:10.1101/gr.074492.107.
URL http://genome.cshlp.org/cgi/content/abstract/18/5/821

[122] S. Altschul, W. Gish, W. Miller, E. Myers, D. Lipman, Basic local alignment search tool., J Mol Biol. 215 (3) (1990) 403–10.

[123] M. Moore, A. Dhingra, P. Soltis, R. Shaw, W. Farmerie, K. Folta, D. Soltis, Rapid and accurate pyrosequencing of angiosperm plastid genomes, BMC Plant Biology 6 (1) (2006) 17. doi:10.1186/1471-2229-6-17.
URL http://www.biomedcentral.com/1471-2229/6/17

[124] S. Goldberg, J. Johnson, D. Busam, T. Feldblyum, S. Ferriera, R. Friedman, A. Halpern, H. Khouri, S. Kravitz, F. Lauro, K. Li, Y. Rogers, R. Strausberg, G. Sutton, L. Tallon, T. Thomas, E. Venter, M. Frazier, J. Venter, A Sanger/pyrosequencing hybrid approach for the generation of high-quality draft assemblies of marine microbial genomes, Proceedings of the National Academy of Sciences 103 (30) (2006) 11240–11245. arXiv:http://www.pnas.org/content/103/30/11240.full.pdf+html, doi:10.1073/pnas.0604351103.
URL http://www.pnas.org/content/103/30/11240.abstract

[125] A. Bashir, S. Volik, C. Collins, V. Bafna, B. Raphael, Evaluation of paired-end sequencing strategies for detection of genome rearrangements in cancer., PLoS Comput Biol 4 (4) (2008) e1000051.

[126] D. Hernandez, P. François, L. Farinelli, M. Osterås, J. Schrenzel, De novo bacterial genome sequencing: millions of very short reads assembled on a desktop computer., Genome Res. 18 (5) (2008) 802–9.

[127] S. Needleman, C. Wunsch, A general method applicable to the search for similarities in the amino acid sequence of two proteins., J Mol Biol. 48 (3) (1970) 443–53.

[128] T. Smith, M. Waterman, Identification of common molecular subsequences., J Mol Biol. 147 (1) (1981) 195–7.

[129] O. Gotoh, An improved algorithm for matching biological sequences., J Mol Biol. 162 (3) (1982) 705–8.

[130] G. Pertea, X. Huang, F. Liang, V. Antonescu, R. Sultana, S. Karamycheva, Y. Lee, J. White, F. Cheung, B. Parvizi, et al., TIGR Gene Indices clustering tools (TGICL): a software system for fast clustering of large EST datasets., Bioinformatics 19 (5) (2003) 651–2.

[131] J. Gorodkin, S. Cirera, J. Hedegaard, M. Gilchrist, F. Panitz, C. Jorgensen, K. Scheibye-Knudsen, T. Arvin, S. Lumholdt, M. Sawera, T. Green, B. Nielsen, J. Havgaard, C. Rosenkilde, J. Wang, H. Li, R. Li, B. Liu, S. Hu, W. Dong, W. Li, J. Yu, J. Wang, H. Staefeldt, R. Wernersson, L. Madsen, B. Thomsen, H. Hornshoj, Z. Bujie, X. Wang, X. Wang, L. Bolund, S. Bruna k, H. Yang, C. Bendixen, M. Fredholm, Porcine transcriptome analysis based on 97 non-normalized cdna libraries and assembly of 1,021,891 ests, Genome Biology 8 (2007) R45.

[132] T. Chen, S. Skiena, A case study in genome-level fragment assembly., Bioinformatics 16 (6) (2000) 494–500.

[133] P. Pevzner, H. Tang, M. Waterman, An Eulerian path approach to DNA fragment assembly., Proc Natl Acad Sci U S A 98 (17) (2001) 9748–53.

[134] T. Scheetz, N. Trivedi, C. Roberts, T. Kucaba, B. Berger, N. Robinson, C. Birkett, A. Gavin, B. O'Leary, T. Braun, et al, ESTprep: preprocessing cDNA sequence reads., Bioinformatics. 19 (11) (2003) 1318–1324.

[135] H. Chou, M. Holmes, DNA sequence quality trimming and vector removal., Bioinformatics 17 (12) (2001) 1093–104.

[136] G. Pertea, seqclean, http://www.tigr.org/tdb/tgi/software/.

[137] K. Schneeberger, K. Malde, E. Coward, I. Jonassen, Masking repeats while clustering ESTs., Nucleic Acids Res. 33 (7) (2005) 2176–80.

[138] A. Smit, R. Hubley, P. Green, RepeatMasker Open-3.0, <http://www.repeatmasker.org> (1996-2004).

[139] J. Jurka, Repbase update: a database and an electronic journal of repetitive elements., Trends Genet. 16 (9) (2000) 418–20.

[140] J. Bedell, I. Korf, W. Gish, MaskerAid: a performance enhancement to RepeatMasker., Bioinformatics. 16 (11) (2000) 1040–1.

[141] J. Quackenbush, F. Liang, I. Holt, G. Pertea, J. Upton, The TIGR gene indices: reconstruction and representation of expressed gene sequences., Nucleic Acids Res 28 (1) (2000) 141–5.

[142] J. Burke, D. Davison, W. Hide, d2_cluster: a validated method for clustering EST and full-length cDNAsequences., Genome Res 9 (11) (1999) 1135–42.

[143] T. Keane, Z. Ning, Assessing Assemblability of Reads from New Sequencing Platforms. ISMB 2007, http://minds.nuim.ie/ tkeane/publications/ismb2007Poster.pdf (2007).

[144] J. Dohm, C. Lottaz, T. Borodina, H. Himmelbauer, Substantial biases in ultra-short read data sets from high-throughput DNA sequencing, Nucl. Acids Res. 36 (16) (2008) e105. arXiv:http://nar.oxfordjournals.org/cgi/reprint/36/16/e105.pdf, doi:10.1093/nar/gkn425.
URL http://nar.oxfordjournals.org/cgi/content/abstract/36/1%6/e105

[145] Y. Erlich, P. Mitra, M. Delabastide, R. Mccombie, G. Hannon, Alta-cyclic: a self-optimizing base caller for next-generation sequencing, Nature Methods 5 (8) (2008) 679–682. doi:http://dx.doi.org/10.1038/nmeth.1230.
URL http://dx.doi.org/10.1038/nmeth.1230

[146] J. Butler, I. MacCallum, M. Kleber, I. Shlyakhter, M. Belmonte, E. Lander, C. Nusbaum, D. Jaffe, Allpaths: de novo assembly of whole-genome shotgun microreads, Genome Res 18 (5) (2008) 810–20.

[147] M. Pop, D. Kosack, S. Salzberg, Hierarchical scaffolding with Bambus., Genome Res. 14 (1) (2004) 149–59.

[148] D. Gordon, C. Abajian, P. Green, Consed: a graphical tool for sequence finishing., Genome Res 8 (3) (1998) 195–202.

[149] D. Gordon, C. Desmarais, P. Green, Automated finishing with autofinish., Genome Res. 11 (4) (2001) 614–25.

[150] D. Bartels, S. Kespohl, S. Albaum, T. Druke, A. Goesmann, J. Herold, O. Kaiser, A. Puhler, F. Pfeiffer, G. Raddatz, et al., BACCardI - a tool for the validation of genomic assemblies, assisting genome finishing and intergenome comparison., Bioinformatics 21 (7) (2005) 853–9.

[151] J. Bonfield, K. Smith, R. Staden, A new DNA sequence assembly program., Nucleic Acids Res. 23 (24) (1995) 4992–9.

[152] E. Mardis, Anticipating the 1,000 dollar genome, Genome Biol 7 (7) (2006) 112.

[153] C. Hutchison, Dna sequencing: bench to bedside and beyond, Nucleic Acids Res 35 (18) (2007) 6227–37.

[154] D. R. Bentley, Whole-genome re-sequencing., Curr Opin Genet Dev 16 (6) (2006) 545–552.

[155] M. Stratton, Genome resequencing and genetic variation, Nature Biotechnology 26 (2008) 65–66.

[156] D. Wheeler, M. Srinivasan, M. Egholm, Y. Shen, L. Chen, A. McGuire, W. He, Y. Chen, V. Makhijani, G. Roth, X. Gomes, K. Tartaro, F. Niazi, C. Turcotte, G. Irzyk, J. Lupski, C. Chinault, X. Song, Y. Liu, Y. Yuan, L. Nazareth, X. Qin, D. Muzny, M. Margulies, G. Weinstock, R. Gibbs, J. Rothberg, The complete genome of an individual by massively parallel dna sequencing, Nature 452 (7189) (2008) 872–876. doi:http://dx.doi.org/10.1038/nature06884.
URL http://dx.doi.org/10.1038/nature06884

[157] J. Wang, G. Wong, P. Ni, Y. Han, X. Huang, J. Zhang, C. Ye, Y. Zhang, J. Hu, K. Zhang, et al., RePS: a sequence assembler that masks exact repeats identified from the shotgun data., Genome Res 12 (5) (2002) 824–31.

[158] M. Taylor, C. Semple, Sushi gets serious: the draft genome sequence of the pufferfish Fugu rubripes., Genome Biol. 3 (9).

[159] P. Havlak, R. Chen, K. Durbin, A. Egan, Y. Ren, X. Song, G. Weinstock, R. Gibbs, The Atlas genome assembly system., Genome Res. 14 (4) (2004) 721–32.

[160] K. Mita, M. Kasahara, S. Sasaki, Y. Nagayasu, T. Yamada, H. Kanamori, N. Namiki, M. Kitagawa, H. Yamashita, Y. Yasukochi, et al., The genome sequence of silkworm, Bombyx mori., DNA Res. 11 (1) (2004) 27–35.

[161] X. Huang, J. Wang, S. Aluru, S. Yang, L. Hillier, PCAP: a whole-genome assembly program., Genome Res. 13 (9) (2003) 2164–70.

[162] G. Sutton, O. White, M. Adams, A. Kerlavage, TIGR Assembler: A New Tool for Assembling Large Shotgun Sequencing Project., Genome Sci Tech 1 (1) (1995) 9–19.

[163] C. Lee, C. Grasso, M. Sharlow, Multiple sequence alignment using partial order graphs., Bioinformatics. 18 (3) (2002) 452–64.

[164] N. Kim, S. Shin, S. Lee, ASmodeler: gene modeling of alternative splicing from genomic alignment of mRNA, EST and protein sequences., Nucleic Acids Res. 32(Web Server issue) (2004) W181–6.

[165] Z. Zhang, S. Schwartz, L. Wagner, W. Miller, A greedy algorithm for aligning DNA sequences., L.

[166] A. Kalyanaraman, S. Aluru, S. Kothari, V. Brendel, Efficient clustering of large EST data sets on parallel computers., Nucleic Acids Res. 31 (11) (2003) 2963–74.

[167] S. Cawley, L. Pachter, HMM sampling and applications to gene finding and alternative splicing., Bioinformatics 19 Suppl 2 (2003) II36–II41.

[168] W. Kent, D. Haussler, Assembly of the working draft of the human genome with GigAssembler., Genome Res. 11 (9) (2001) 1541–8.

[169] Mouse Genome Sequencing Consortium, Initial sequencing and comparative analysis of the mouse genome., Nature 420 (6915) (2002) 520–62.

[170] G. Marth, I. Korf, M. Yandell, R. Yeh, Z. Gu, H. Zakeri, N. Stitziel, L. Hillier, P. Kwok, W. Gish, A general approach to single-nucleotide polymorphism discovery., Nat Genet. 23 (4) (1999) 452–6.

[171] D. Nickerson, V. Tobe, S. Taylor, PolyPhred: automating the detection and genotyping of single nucleotide substitutions using fluorescence-based resequencing., Nucleic Acids Res 25 (14) (1997) 2745–51.

[172] S. Weckx, J. Del Favero, R. Rademakers, L. Claes, M. Cruts, J. P. De, B. C. Van, R. P. De, novoSNP, a novel computational tool for sequence variation discovery., Genome Res. 15 (3) (2005) 436–42.

[173] V. Ramensky, P. Bork, S. Sunyaev, Human non-synonymous SNPs: server and survey., Nucleic Acids Res. 30 (17) (2002) 3894–900.

[174] P. Ng, S. Henikoff, SIFT: Predicting amino acid changes that affect protein function., Nucleic Acids Res. 31 (13) (2003) 3812–4.

[175] B. Modrek, A. Resch, C. Grasso, C. Lee, Genome-wide detection of alternative splicing in expressed sequences of human genes., Nucleic Acids Res. 29 (13) (2001) 2850–9.

[176] J. Bouck, W. Yu, R. Gibbs, K. Worley, Comparison of gene indexing databases., Trends Genet. 15 (4) (1999) 159–62.

[177] F. Chen, E. Vallender, H. Wang, C. Tzeng, W. Li, Genomic divergence between human and chimpanzee estimated from large-scale alignments of genomic sequences., J Hered 92 (6) (2001) 481–489.

[178] J. Venter, K. Remington, J. Heidelberg, A. Halpern, D. Rusch, J. Eisen, D. Wu, I. Paulsen, K. Nelson, W. Nelson, et al, Environmental genome shotgun sequencing of the Sargasso Sea., Science 2004.

[179] G. Tyson, J. Chapman, P. Hugenholtz, E. Allen, R. Ram, P. Richardson, V. Solovyev, E. Rubin, D. Rokhsar, J. Banfield, Community structure and metabolism through reconstruction of microbial genomes from the environment., Nature.

[180] S. Tringe, C. von Mering, A. Kobayashi, A. Salamov, K. Chen, H. Chang, M. Podar, J. Short, E. Mathur, J. Detter, et al, Comparative metagenomics of microbial communities., Science 308 (5721) (2005) 554–557.

[181] H. Poinar, C. Schwarz, J. Qi, B. Shapiro, R. Macphee, B. Buigues, A. Tikhonov, D. Huson, L. Tomsho, A. Auch, M. Rampp, W. Miller, S. Schuster, Metagenomics to paleogenomics: large-scale sequencing of mammoth dna, Science 311 (5759) (2006) 392–4.

[182] S. Seemann, M. Gilchrist, I. Hofacker, P. Stadler, J. Gorodkin, Detection of RNA structures in porcine est data and related mammals, BMC Genomics 8 (2007) 316.

[183] D. Jaffe, J. Butler, S. Gnerre, E. Mauceli, K. Lindblad Toh, J. Mesirov, M. Zody, E. Lander, Whole-genome sequence assembly for mammalian genomes: Arachne 2., Genome Res 13 (1) (2003) 91–6.

[184] J. Dohm, C. Lottaz, T. Borodina, H. Himmelbauer, SHARCGS, a fast and highly accurate short-read assembly algorithm for de novo genomic sequencing, Genome Res 17 (11) (2007) 1697–706.

[185] R. Warren, G. Sutton, S. Jones, R. Holt, Assembling millions of short dna sequences using ssake., Bioinformatics 23 (4) (2007) 500–1.

[186] Y. Xing, A. Resch, C. Lee, The multiassembly problem: reconstructing multiple transcript isoforms from EST fragment mixtures., Genome Res. 14 (3) (2004) 426–41.

Figures

Figure 1 – Timeline

Figure 1: Figure showing the major breakthroughs in sequencing. The year of the different milestones is chosen to be the publication year of the first article that presented the method. Software publications are marked in cursive. On the left, the size of GenBank (in deposited basepairs) is shown, with the length and width of the bars representing the size on a logarithmic scale.

Figure 2 – Sequencing vector

Figure 2: Figure showing a schematic drawing of a sequencing vector, such as a BAC (Bacterial Artificial Chromosome). The insert can be a genomic fragment, or an cDNA (for EST sequencing). In both cases sequencing from each end will produce a read pair that can provide additional information for assemblers.

Figure 3 – Sequencing methods

Figure 3: Schematic drawing of the four different sequencing procedures. (a) Hierarchical shotgun, where a BAC clone map (tilling map) covering the genome is first created after which the BACs are sequenced. (a) Whole Genome Shotgun, where the genome is randomly split into smaller parts and sequenced. (c) EST sequencing, where mRNA is extracted from tissue and then sequenced. (d) Massively parallel sequencing where short sequence fragments are aligned to a reference genome.

Figure 4 – Repeat Contig

Figure 4: Schematic drawing of a cluster contain a likely repeat. The region on the right is covered by many more reads than would be expected by chance, and is therefore potentially a repeat region, which could be masked.

Figure 5 – Assembly pipeline

Figure 5: Figure showing the typical pipeline of a sequencing project. Sequenced reads are generated, after which they are cleaned and assembled. Following the assembly annotation and analysis can be performed.The grey line show the pipeline for massively parallel sequencing where the reads are mapped to a reference genome, while the full pipeline is for de novo sequencing ans assembly. Part of the figure is adapted from [182]

Figure 6 – Assembly example

Figure 6: Figure showing an examples of an assembled (EST) contig (cluster). The thick line at the top represents the consensus sequence produced by the applied assembler ([131]). The blowup shows a putative SNP present in the sequences. The colored stretches mark specific tri-nucleotides, 'ATG' is green and 'TAA' is red, and are drawn to show the structure of the assembly.

Figure 7 – Graph example

Figure 7: Figure showing an examples of how a graph is constructed. Two reads are mapped onto the different k-mer nodes ($k = 6$ in this example), and edges between the nodes are determined by the reads. The presence of a nucleotide difference (*eg.* sequencing error, SNP, *etc*.) between the two reads cause the graph to split up, thus causing an ambiguity in the sequence.

Fig. 1

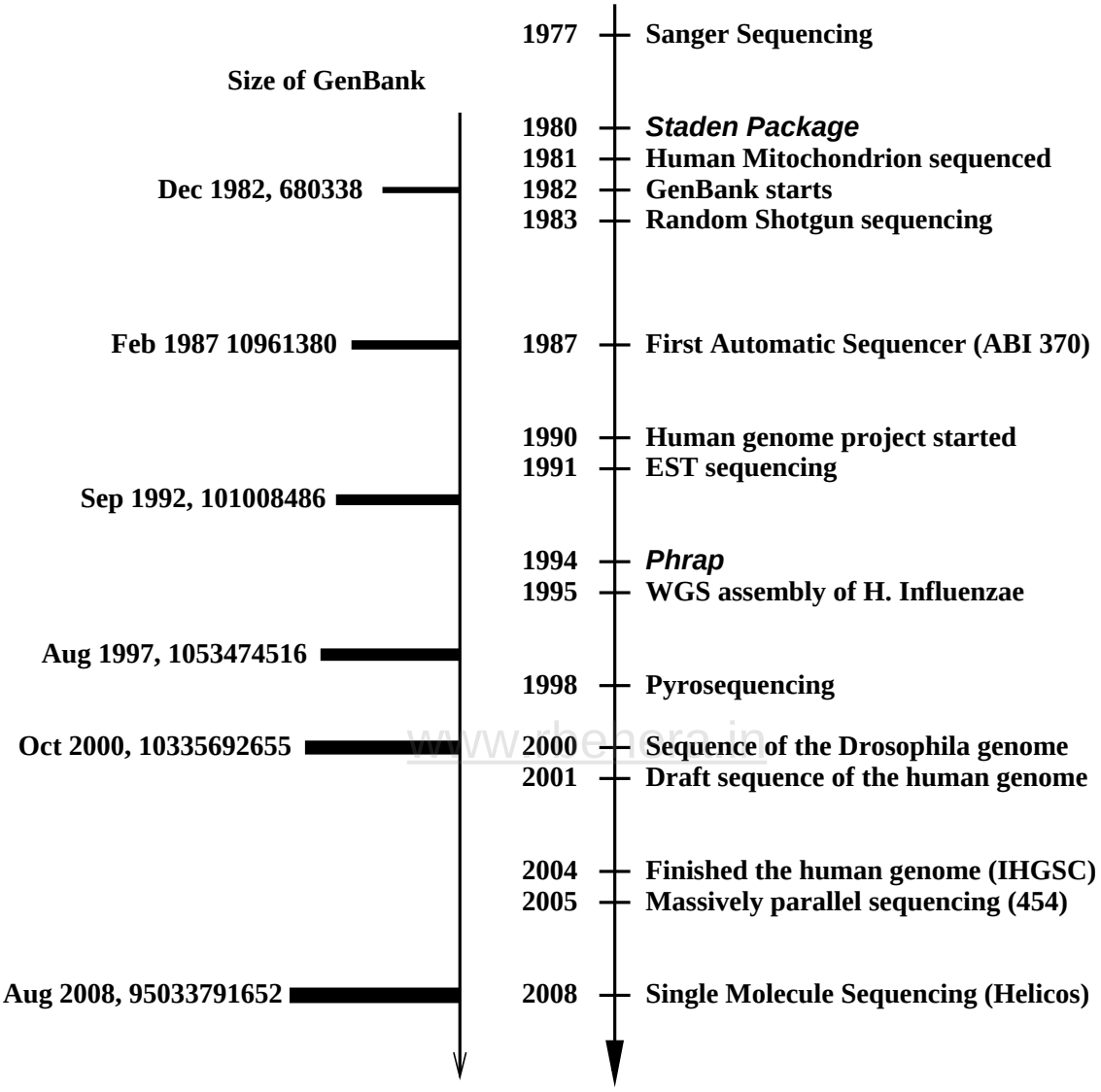
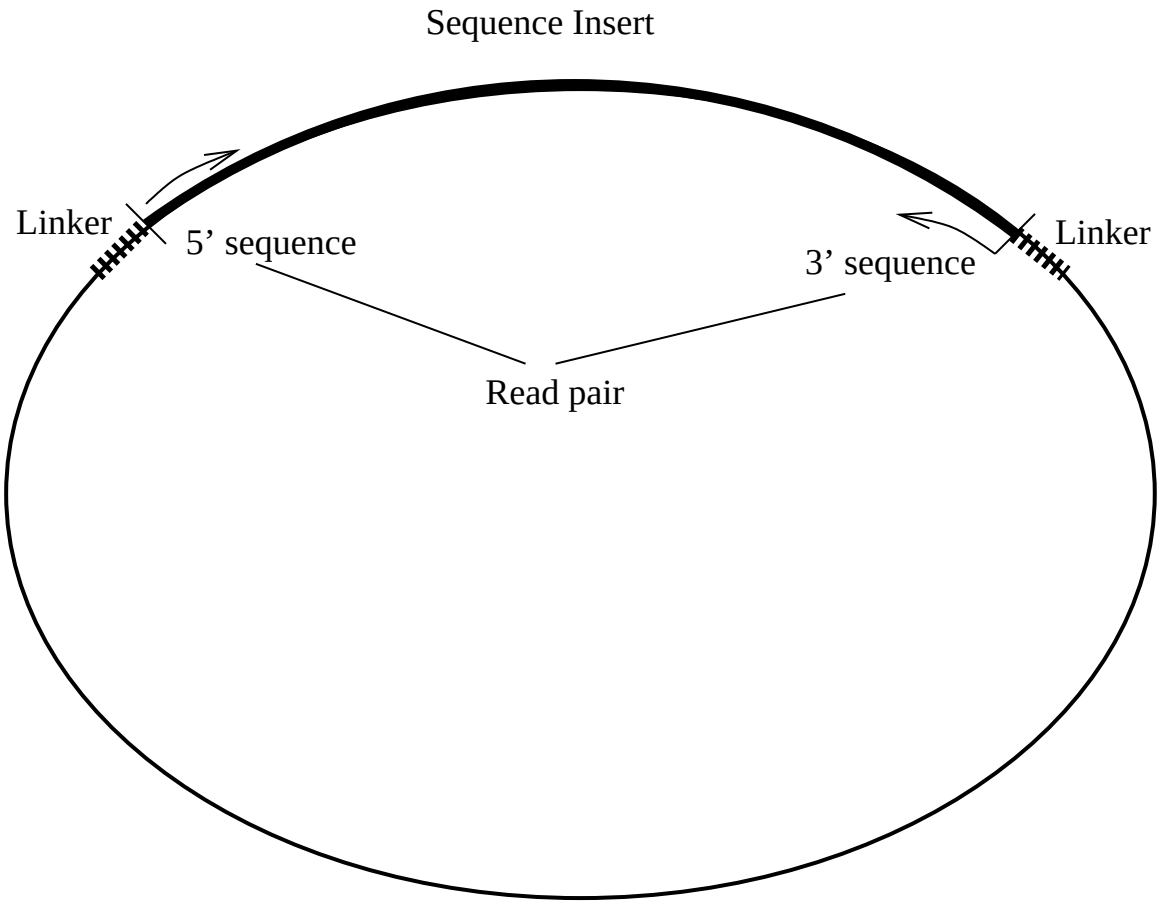


Fig. 2



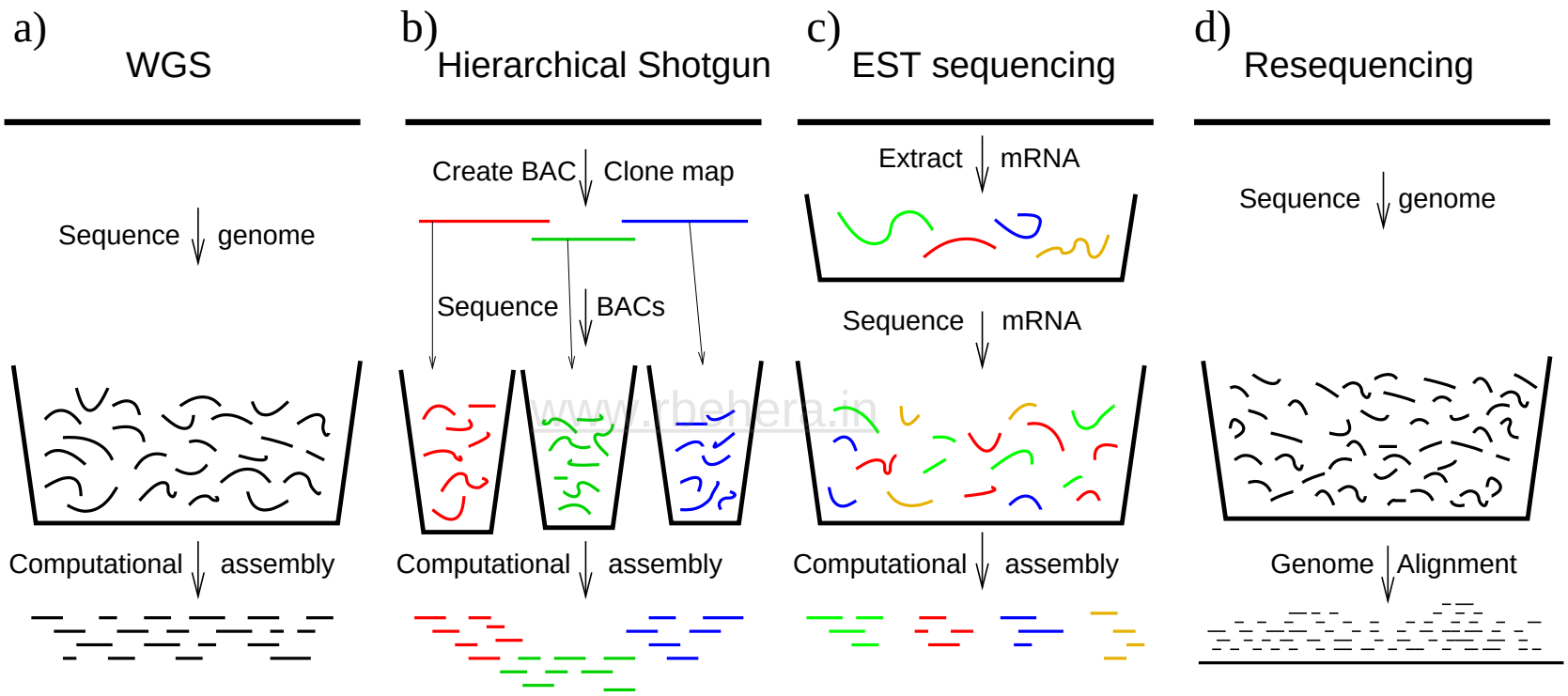


Fig. 3

Fig. 4

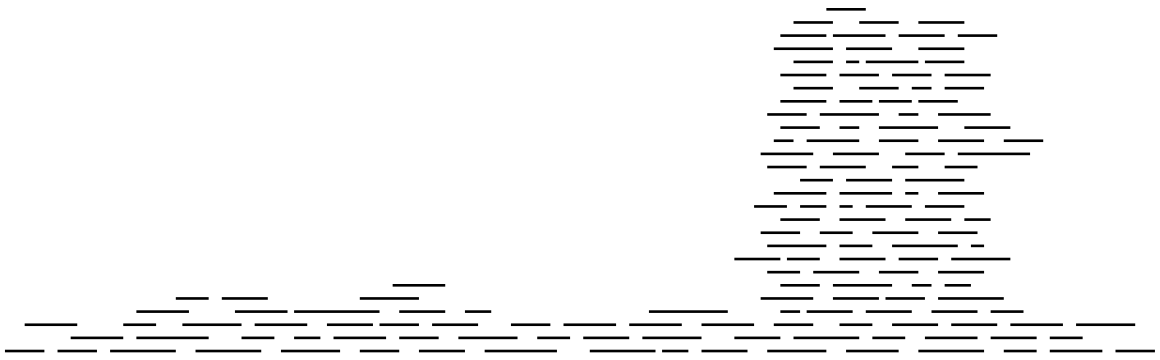


Fig. 5

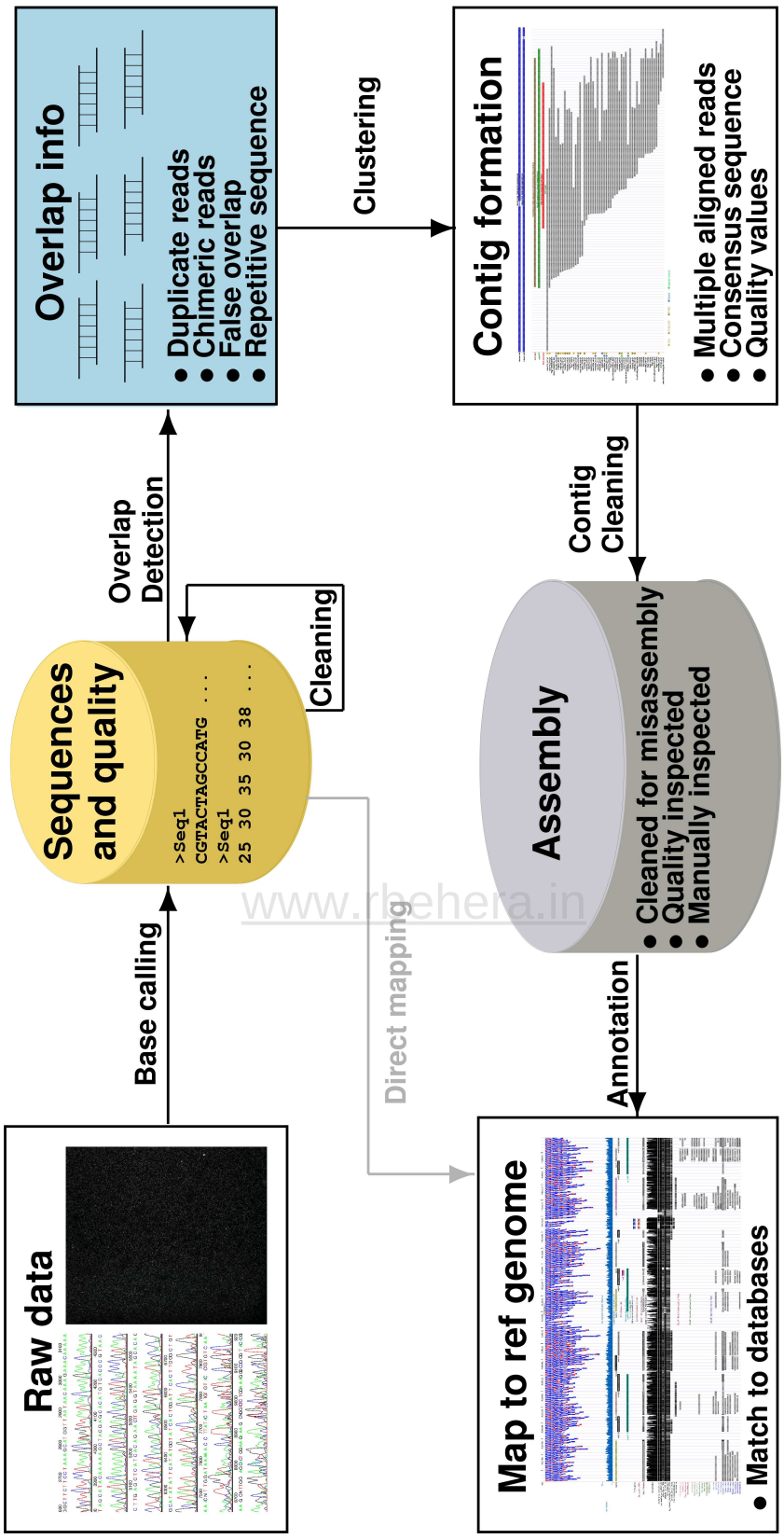


Fig. 6

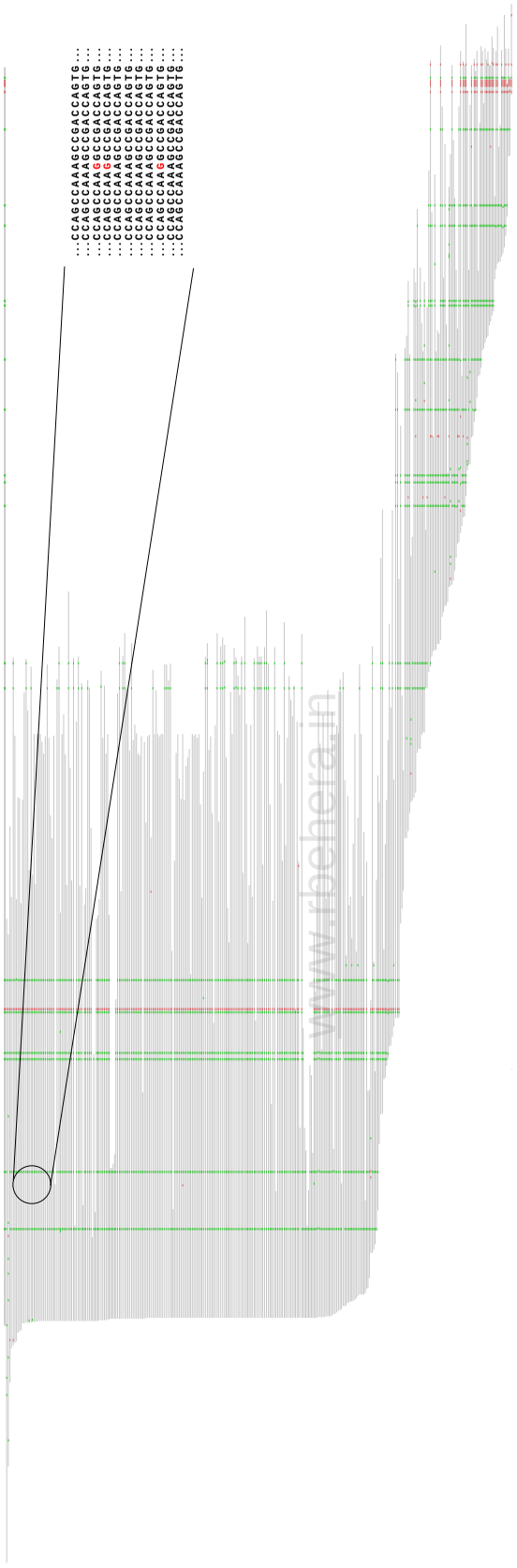
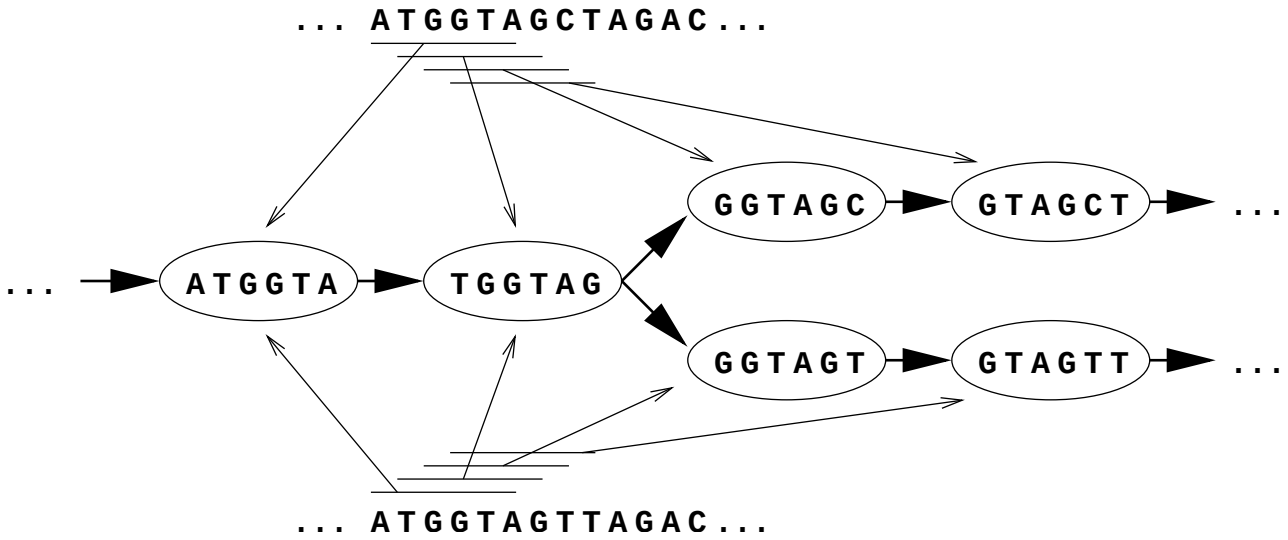


Fig. 7



Tables

Table 1a – Assemblers used primarily for shotgun data.

Assembler	Computational dependencies	Additional Information	Common Features	Reference
Phusion	RPPHRAP	PR, BAC,Q_r	P, R, K	[50]
JAZZ	banded SW, malign, PHRAP	Q_r	K	[158]
RePS	BLAST/PHRAP	PR	R, K	[157]
ARACHNE2	SW	Q_r, PR	K	[49] [183]
GigAssembler	psLayout	PR, BAC, EST, Q_r	P, R	[168]
Celera assembler	BLAST-like	PR	P	[45]
Euler	graph-based	PR	R	[133]
CAP3	banded SW	Q_r, PR	P	[116]
GAP4	CAP3, PHRAP or FAKII	Q, PR		[151]
RAMEN	banded SW	Q_r	R	[160]
ATLAS	PHRAP, banded SW	Q_r	R, K	[159]
PCAP	CAP3, banded SW		P, R	[161]
Bambus	-	contigs	P	[147]
TRAP	mod SW	Q_r	R, K	[58]
PHRAP	banded SW	Q		[44]
TIGR Assembler	mod SW	Q	R	[162]
STROLL	banded SW	Q		[132]
mira	banded SW	Q_r	R	[59]
ALLPATHS	graph-based	PR		[146]
SHARCGS	contig elongation			[184]
Velvet	graph-based	PR		[121]
SSAKE	contig elongation			[185]

Table 1a: Overview of different assembly programs (including scaffolders), some of the programs have also been used to assemble EST sequences. The additional information shows the information which a given assembler can use, besides read information. **PR**: Paired Reads information, **BAC**: Bacterial artificial Chromosome data, **Q**: quality data, **Q_r**: Quality data and trimming reads without sufficient quality. Common features are features that the assembler shares with other assemblers: **P**: Process can be run on parallel computers, **R**: Handles repeats, **K**: K-mer approach to find potential overlaps. The last four programs listed are designed primarily for short read assembly.

Table 1b – "Assemblers" designed for ESTs

Program	Computational dependencies	Additional Information	Common Features	Reference
TGICL	megablast/CAP3	known genes, Q_r	P	[141]
StackPack	PHRAP	Q_r		[13]
PaCE	Suffix tree		R	[166]
Splicing graphs	graph-based			[55]
ASmodeler	Directed acyclic graph	mRNA, EST protein sequences		[164]
HB-algorithm	HB-algorithm	EST		[186]
geneDistiller	megablast	Q_r		[54]
xtract	graph-based	Q_r		[57]

Table 1b: Overview of the programs designed for clustering, analysis and assembly of EST data. See table 1a for abbreviations.

Pairwise Sequence alignment

♦ Pairwise sequence alignment methods are used to find the best-matching local or global alignments of two query sequences.

♦ Pairwise alignments can only be used between two sequences at a time.

♦ It is used to identify regions of similarity that may indicate functional, structural and/or evolutionary relationships between two biological sequences (protein or nucleic acid).

```
A: C A T - T C A - C
   |   |     | |   |
B: C - T C G C A G C
```

In global alignment, two sequences to be aligned are assumed to be generally similar over their entire length.

Alignment is carried out from beginning to end of both sequences to find the best possible alignment across the entire length between the two sequences.

Local alignment, on the other hand, does not assume that the two sequences in question have similarity over the entire length.

It only finds local regions with the highest level of similarity between the two sequences and aligns these regions without regard for the alignment of the rest of the sequence regions.

S = CTGTCGCTGCACG
T = TGCCGTG

The three primary methods of producing **pairwise alignments** are **dot-matrix methods**, **dynamic programming**, and **word methods**.

Sequence Alignment (Needleman-Wunsch, Smith-Waterman)

Topics:

1. **Needleman-Wunsch** (Global Alignment)
2. **Maximum Contiguous Subsequence Sum** (Not Required For BSc. Biotech)
3. **Smith-Waterman** (Local Alignment)

Background: Importance of Sequence Alignment
Comparative analysis is the backbone of evolutionary biology. It was phenotypic variation which allowed Darwin to compose his theory of natural selection. That theory rests on the fact that transfer of the genetic code from parent to progeny does not exist without change. It is these changes in genetic sequence which allow for divergence of species, and thus provide a backdrop for natural selection. Just as comparative analysis was key for evolutionary biology, sequence alignment is the cornerstone of modern bioinformatics. Rapid and automated sequence analysis facilitates everything from functional classification & structural determination of proteins, to studies of genetic expression and evolution.

1. Needleman-Wunsch (Global Alignment)

Dynamic programming algorithms find the best solution by breaking the original problem into smaller sub-problems and then solving. The Needleman-Wunsch algorithm is a dynamic programming algorithm for optimal sequence alignment (Needleman and Wunsch, 1970). Basically, the concept behind the Needleman-Wunsch algorithm stems from the observation that any partial sub-path that tends at a point along the true optimal path must itself be the optimal path leading up to that point. Therefore the optimal path can be determined by incremental extension of the optimal sub-paths. In a Needleman-Wunsch alignment, the optimal path must stretch from beginning to end in both sequences (hence the term 'global alignment').

In order to perform a Needleman-Wunsch alignment, a matrix is created which allows us to compare the two sequences. The score $M(i,j)$ for every cell depends on the three cells corresponding to either or both sequence having 1 less letter (i.e. cells $M(i-1.j)$, $M(i,j-1)$ and $M(i-1,j-1)$. It is calculated as follows:

$$M(i,j) = \mathrm{MAX}(M_{i-1,j-1} + S(A_i, B_j)$$
$$M_{i-1,\,j} + \mathrm{gap}$$
$$M_{i,j-1} + \mathrm{gap})$$

where gap is the gap penalty and the function S returns the score/penalty for matching the two corresponding letters. Once we have computed this score for every cell, we must do a "traceback", that is to determine the actual set of operations that lead to the score.

Because when computing the score of a cell we took a max over three numbers, on the traceback we go to the location of the highest – going sideways or up corresponds to gaps, and going along the diagonal corresponds to a match. This algorithm performs alignments with a time complexity of O(mn) and a space complexity of O(mn).

Example:

Find the best alignment of these two sequences:

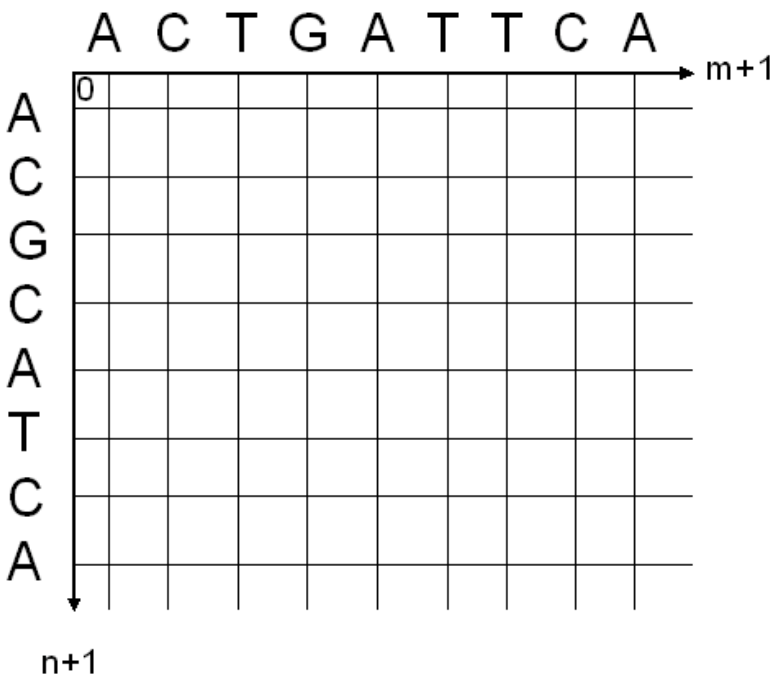
ACTGATTCA
ACGCATCA

Using -2 as a gap penalty, -3 as a mismatch penalty, and 2 as the score for a match.

Solution:

Step 1: Draw the matrix

For 2 sequences (length m and length n) what size scoring matrix is needed for their alignment? Grid dimensions must be (m+1) × (n+1). Think of each increment as a division of the sequence members:



Step 2: Assign scores

		A	C	T	G	A	T	T	C	A
	0	-2	-4	-6	-8	-10	-12	-14	-16	-18
A	-2	2	0	-2	-4	-6	-8	-10	-12	-14
C	-4	0	4	2	0	-2	-4	-6	-8	-10
G	-6	-2	2	1	4	2	0	-2	-4	-6
C	-8	-4	0	-1	2	1	-1	-3	0	-2
A	-10	-6	-2	-3	0	4	2	0	-2	2
T	-12	-8	-4	0	-2	2	6	4	2	0
C	-14	-10	-6	-2	-4	0	4	2	6	4
A	-16	-12	-8	-4	-5	-2	2	1	4	8

Step 3: Trace back

The optimal path is traced beginning from the lower right-hand corner

		A	C	T	G	A	T	T	C	A
	(0)	-2	-4	-6	-8	-10	-12	-14	-16	-18
A	-2	(2)	0	-2	-4	-6	-8	-10	-12	-14
C	-4	0	(4)	(2)	0	-2	-4	-6	-8	-10
G	-6	-2	2	1	(4)	2	0	-2	-4	-6
C	-8	-4	0	-1	(2)	1	-1	-3	0	-2
A	-10	-6	-2	-3	0	(4)	2	0	-2	2
T	-12	-8	-4	0	-2	2	(6)	(4)	2	0
C	-14	-10	-6	-2	-4	0	4	2	(6)	4
A	-16	-12	-8	-4	-5	-2	2	1	4	(8)

Result:

This analysis yielded the following alignment:

```
ACTG-ATTCA
||   || ||
AC-GCAT-CA
```

The alignment score is equal to the value in the lower right-hand corner of the matrix (8).

2. From Global to Local Similarity: Maximum Contiguous Subsequence Sum

When aligning two very large sequences, it is often useful to determine the locations of high similarity regions, even if there is no additional similarity inbetween the sequences. Now that we know how to calculate the *global* alignments, how can we find all local high-scoring hits, or *local* alignments above a given threshold for two large sequences? The answer is related to a programming "pearl", the 'Maximum Contiguous Subsequence Sum' (MSS).

Problem:

Given integers $A_1, A_2, ..., A_N$ find (and identify the sequence corresponding to) the maximum value of:

$$\sum_{k=1}^{j} A_k$$

Solution:

Can be solved in time complexity of 'n'.

```
mss(A) {
      max = 0;
      sum = 0;
      for (i=1; i ≤ n; i+1) {
            sum = sum + A[i];
            if (sum > max)
                  max = sum;
            if (sum < 0)
                  sum = 0;
```

```
        }
        return max;
    }
```

Analysis:

When a subsequence occurs which has a negative sum, the subsequence which will be examined next can begin after the first subsequence (the one that produced the negative sum). Basically, the entire first subsequence is regarded as not having a starting point which will generate a positive sum. For example, consider this set of numbers:

4, 6, -2, 2, -14, 9

Some sums are positive (4, 4+6, 4+6+(-2), 4+6+(-2)+2) but the sum of the first 5 terms (4+6+(-2)+2-14) is negative. Therefore it follows logically that any sequence starting between the 4 and -14 and ending with the -14 will have a negative sum.

The maximum contiguous subsequence sum searches exactly for the highest scoring local area. We now generalize this approach for sequence alignment; the only change is we do the abovealgorithm in two dimensions!

3. Smith-Waterman (Local Alignment)

Over a decade after the initial publication of the Needleman-Wunsch algorithm, a modification was made to allow for local alignments (Smith and Waterman, 1981). In this adaptation, the alignment path does not need to reach the edges of the search graph, but may begin and end internally. In order to accomplish this, 0 was added as a term in the score calculation described by Needleman and Wunsch.

Recall that for global alignments the value at any point is:

$$M(i,j) = \mathrm{MAX}(M_{i-1,j-1} + S(A_i, B_j),\ M_{i-1,j} + \mathrm{gap},\ M_{i,j-1} + \mathrm{gap})$$

However for local alignments:

$$M(I,j) = \mathrm{MAX}(M_{i-1,j-1} + S(A_i, B_j),\ M_{i-1,j} + \mathrm{gap},\ M_{i,j-1} + \mathrm{gap},\ 0)$$

The implication of this is that there are no values below zero in a local alignment scoring matrix, and the reason for the zero is exactly the same as in the MSS problem above.

Example:

Find the best local alignment between these two sequences:

ATGCATCCCATGAC
TCTATATCCGT

Using -2 as a gap penalty, -3 as a mismatch penalty, and 2 as the score for a match.

Solution:

Traceback begins at the highest value (which is also the alignment score).

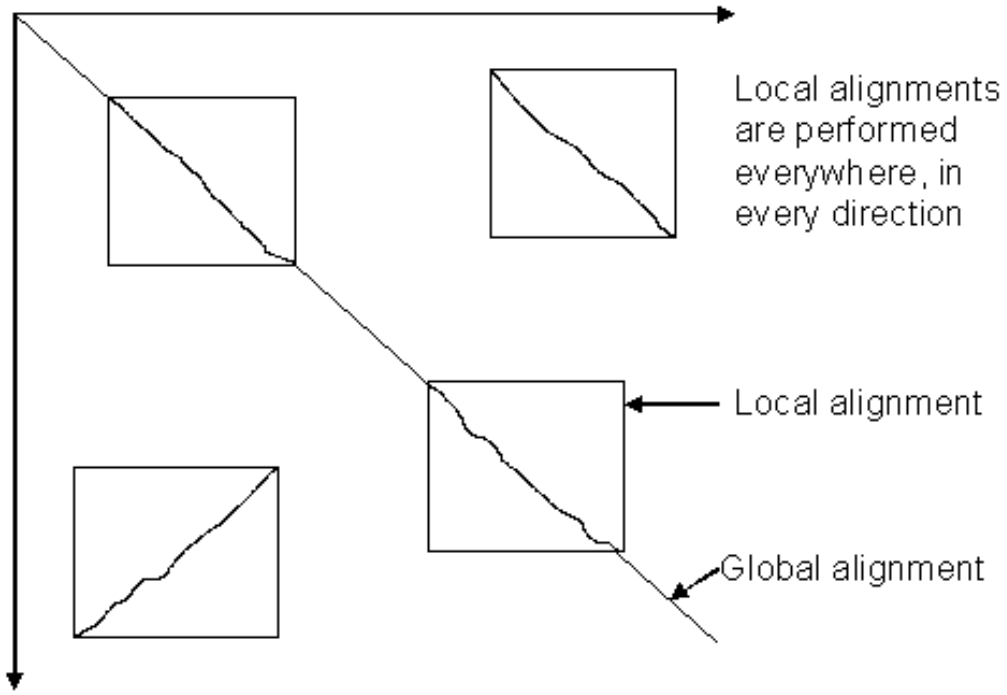
		A	T	G	C	A	T	C	C	C	A	T	G	A	C
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T	0	0	2	0	0	0	2	0	0	0	0	2	0	0	0
C	0	0	0	0	2	0	0	4	2	2	0	0	0	0	2
T	0	0	2	0	0	0	0	2	1	0	0	2	0	0	0
A	0	2	0	0	0	2	0	0	0	0	2	0	0	2	0
T	0	0	4	2	0	0	2	0	0	0	0	4	2	0	0
A	0	2	0	0	0	2	0	0	0	0	2	0	0	2	0
T	0	0	4	2	0	0	4	2	0	0	0	4	0	0	0
C	0	0	2	0	4	0	0	6	4	2	0	0	0	0	2
C	0	0	0	0	2	0	0	4	8	6	4	2	0	0	2
G	0	0	0	2	0	0	0	2	6	5	3	1	4	2	0
T	0	0	2	0	0	0	2	0	4	3	2	5	3	1	0

Which yields the alignment:

ATCC
| | | |
ATCC

With an alignment score of 8.

Local alignments are performed everywhere possible along two sequences.



When trying to find the best local alignments corresponding to a global alignment, a sub-matrix is created with the highest positive score for all alignments above a given threshold. Therefore, the same thing that the MSS was doing on a linear matrix, the Smith-Waterman alignment does on a rectangular matrix.

DIFFERENCES:

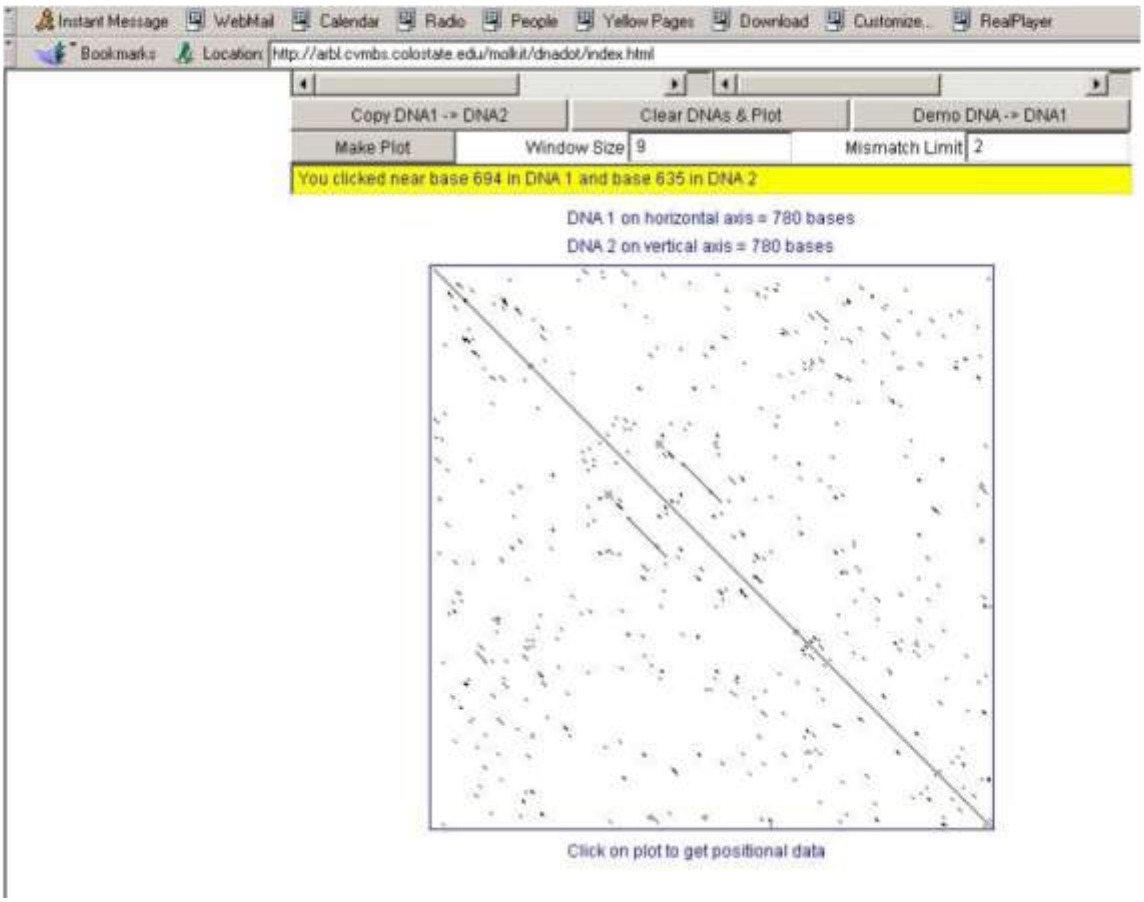
	NEEDLEMAN AND WUNSCH	**SMITH-WATERMAN**
1	Global alignment	Local alignment
2	Requires alignment scores for a pair of residues to be >=0	Requires alignment score for may be positive or negative.
3	No gap penalty required	Requires a gap penalty to work effectively.
4	Score cannot decrease between two cells of a pathway	Score cannot increase, decrease or stay level between two cells of a pathway.

Dot matrix analysis

- A dot matrix is a grid system where the similar nucleotides of two DNA sequences are represented as dots.
- It also called dot plots.
- It is a pairwise sequence alignment made in the computer.
- The dots appear as colourless dots in the computer screen.
- In dot matrix , nucleotides of one sequence are written from the left to right on the top row and those of the other sequence are written from the top to bottom on the left side (column) of the matrix.At every point, where the two nucleotides are the same , a dot in the intersection of row and column becomes a dark dot. when all these darken dots are connected, it gives a graph called dot plot. the line found in the dot plot is called recurrence plot. Each dot in the plot represents a matching nucleotide or amino acid.
- Dot matrix method is a qualitative and simple to analyze sequences.however ,it takes much time to analyze large sequences.
- Dot matrix method is useful for the following studies :
- Sequence similarity between two nucleotide sequences or two amino acid sequences.
- Insertion of short stretches in DNA or amino acid sequence.
- Deletion of short stretches from a DNA or amino acid sequence.
- Repeats or inserted repeats in a DNA or amino acid sequence.

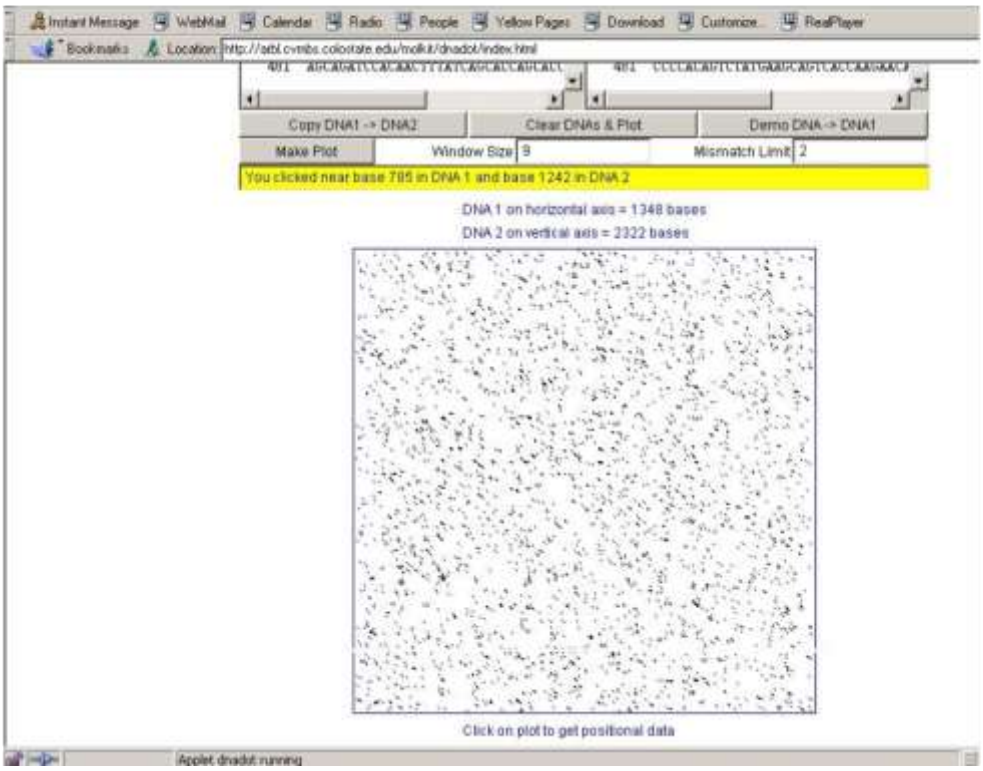
Dot matrix analysis: Two identical sequences

- Nucleic Acids Dot Plots



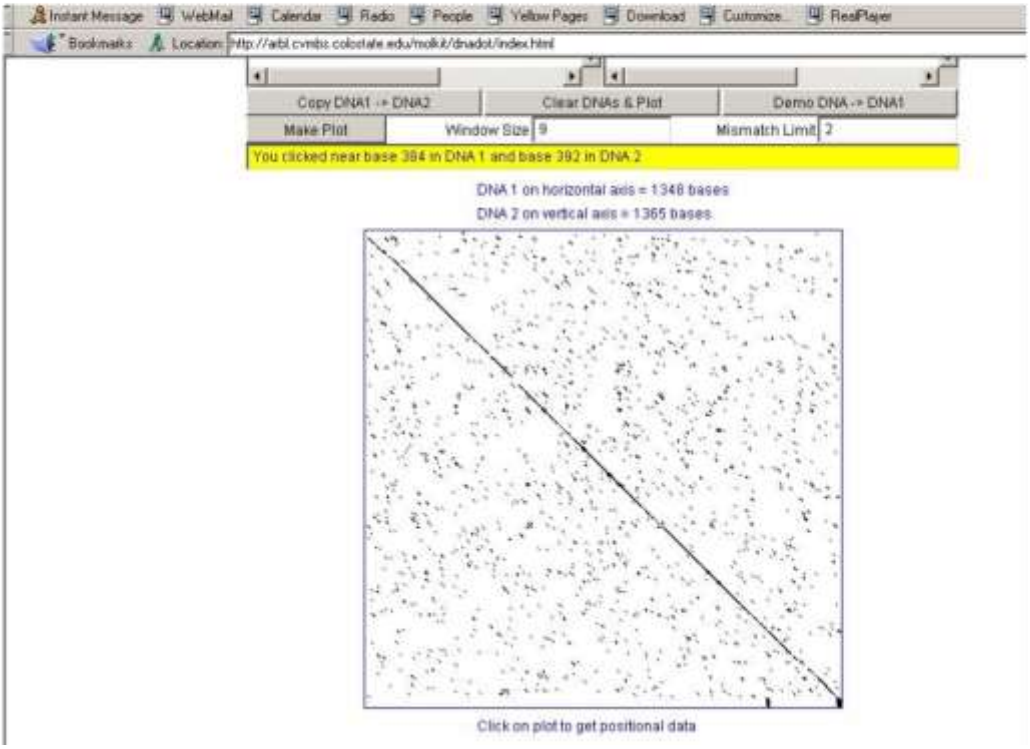
Dot matrix analysis: two very different sequences

- Nucleic Acids Dot Plots of genes



Dot matrix analysis: two similar sequences

- Nucleic Acids Dot Plots of genes



Word Method or K-tuple method

- It is used to find an optimal alignment solution,but is more than dynamic programming .
- This method is useful in large-scale database searches to find whether there is significant match available with the query sequence.
- Word method is used in the database search tools FASTA and the BLAST family .
- They identify a series of short ,non-overlapping subsequences (words) of the query sequence.
- Then they are matched to candidate database sequences to get result .
- In the FASTA method ,the user defines a value *k*to use as the word length to search the database .it is slower but more sensitive at lower values of *k*. they are also perferred for serches involving a very short qurery sequence .
- The BLAST provides a number of algorithms optimized for particular types of queries ,for distantly related sequence matches.
- It is a good alternative to FASTA .However , the results are not very accurate .
- Like FASTA ,BLAST uses a word search of length *k*,but evaluates only the most significant word m,latches rather than every word match .

Multiple Sequence Alignment

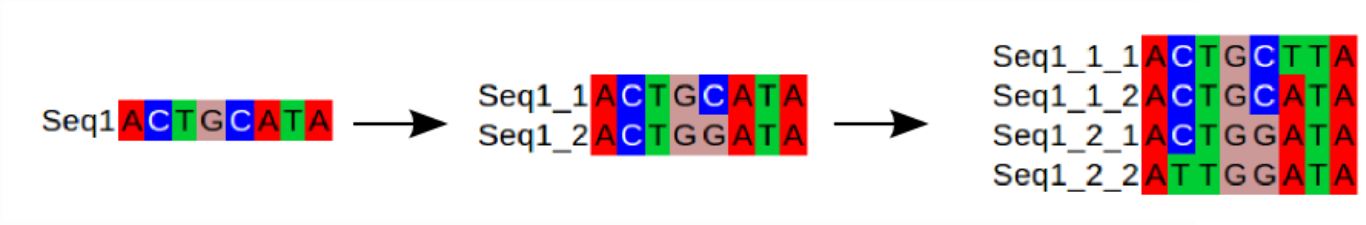
Introduction

A Multiple Sequence Alignment is an alignment of more than two sequences. We could align several DNA or protein sequences.

Some of the most usual uses of the multiple alignments are:

- phylogenetic analysis
- conserved domains
- protein structure comparison and prediction
- conserved regions in promoteres

The multiple sequence alignment asumes that the sequences are homologous, they descend from a common ancestor. The algorithms will try to align homologous positions or regions with the same structure or function.



Multiple alignment algorithm

Multiple alignments are computationally much more difficult than pair-wise alignments. It would be ideal to use an analog of the Smith & Waterman algorithm capable of looking for optimal alignments in the diagonals of a multidimensional matrix given a scoring schema. This algorithm would had to create a multidimensional matrix with one dimension for each sequence. The memory and time required for solving the problem would increase geometrically with the lenght of every sequence. Given the number of sequences usually involved no algorithm is capable of doing that. Every algorithm available reverts to a heuristic capable of solving the problem in a much faster time. The drawback is that the result might not be optimal.

Usually the multiple sequence algorithms assume that the sequences are similar in all its length and they behave like global alignment algorithms. They also assume that thre are not many long insertions and delections. Thus the algorithms will work for some sequences, but not for others.

These algorithms can deal with sequences that are quite different, but, as in the pair-wise case, when the sequences are very different they might have problems creating good algorithm. A good algorithm should align the homologous positions or the positions with the same structure or function.

It we are trying to align two homologous proteins from two species that are phylogenetically very distant we might align quite easily the more conserved regions, like the conserved domains, but we will have problems aligning the more different regions. This was also the case in the pair-wise case, but remember that the multiple alignment algorithms are not guaranteed to give back the best possible alignment.

These algorithms are not design to align sequences that do not cover the whole region, like the reads from a sequencing project. There are other algorithms to assemble sequencing projects.

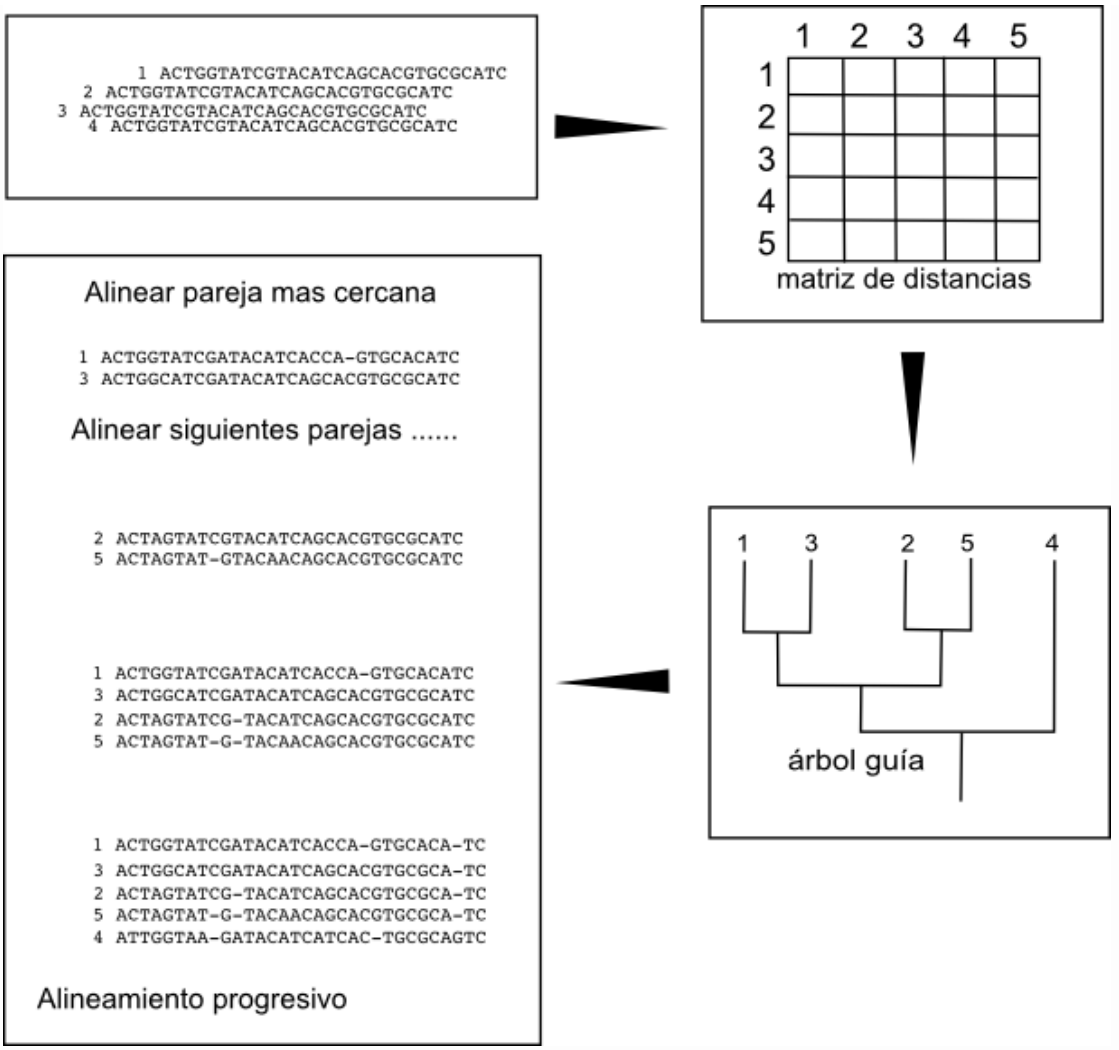
Progressive contruction algorithms

In Multiple Sequence Alignment it is quite common that the algorithms use a progressive alignment strategy. These methods are fast and allow to align thousands of sequences.

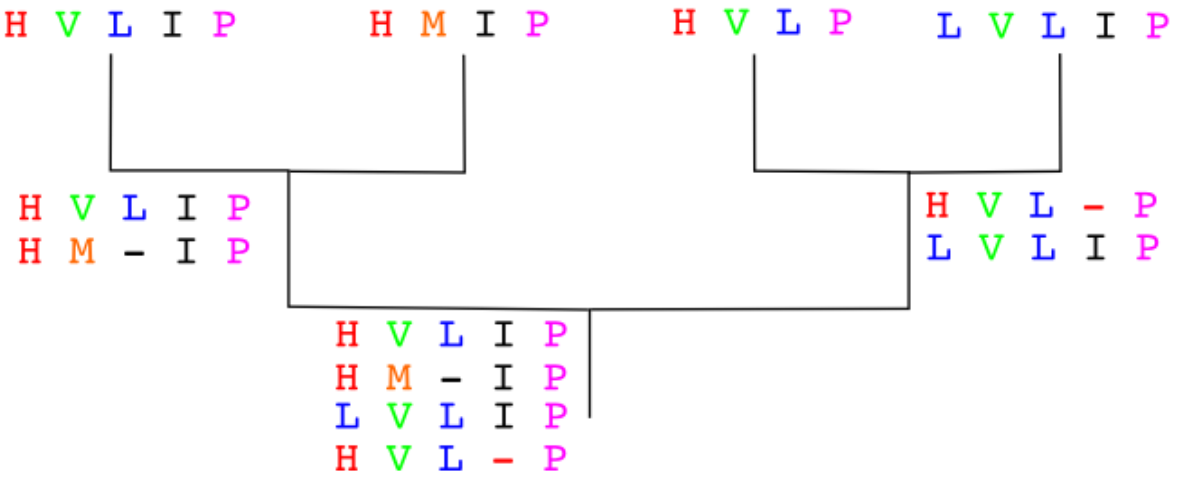
Before starting the alignemnt, as in the pair-wise case, we have to decide which is the scoring schema that we are going to use for the matches, gaps and gap extensions. The aim of the alignment would be to get the multiple sequence alignment with the highest score possible. In the multiple alignment case we do not have any practical algorithm that guarantees that it going to get the optimal solution, but we hope that the solution will be close enough if the sequences comply with the restrictions assumed by the algorithm.

The idea behind the progressive construction algorithm is to build the pair-wise alignments of the more closely related sequences, that should be easier to build, and to align progressively these alignments once we have them. To do it we need first to determine which are the closest sequence pairs. One rough and fast way of determining which are the closest sequence pairs is to align all the possible pairs and look at the scores of those alignments. The pair-wise alignments with the highest scores should be the ones between the more similar sequences. So the first step in the algorithm is to create all the pair-wise alignments and to create a matrix with the scores between the pairs. These matrix will include the similarity relations between all sequences.

Once we have this matrix we can determine the hierarchical relation between the sequences, which are the closest pairs and how those pairs are related and so on, by creating a hierarchical clustering, a tree. We can create these threes by using different fast algorithms like UPGMA or Neighbor joining. These trees are usually known as guide trees.



An example:



Another example:

```
Secuences:
IMPRESIONANTE
INCUESTIONABLE
IMPRESO

Scores:

    IMPRESIONANTE X IMPRESO 7/13

    IMPRESIONANTE X INCUESTIONABLE 10/14

    INCUESTIONABLE X IMPRESO 4/14

Scoring pair-wise matrix:

                IMPRESIONANTE   INCUESTIONABLE   IMPRESO
IMPRESIONANTE         1              10/14          7/13
INCUESTIONABLE      10/14              1            4/14
IMPRESO              7/13             4/14           1

Guide Tree:

       |--- IMPRESIONANTE
   |---|--- INCUESTIONABLE
   |
   |----- IMPRESO

The first alignment would be: IMPRESIONANTE x INCUESTIONABLE

    IMPRES-IONABLE
    INCUESTIANABLE

Now we align IMPRESO to the previous alignment.

    IMPRES-IONANTE
    INCUESTIONABLE
    IMPRES--O-----
```


We have no guarantee that the final is the one with the highest score.

The main problem of these progressive alignment algorithms is that the errors introduced at any point in the process are not revised in the following phases to speed up the process. For instance, if we introduce one gap in the first pair-wise alignment this gap will be propagated to all the following alingments. If the gap was correct that is fine, but if it was not optimal it won't be fixed. These methods are specially prone to fail when the sequences are very different or phylogenetically distant.

```
Sequences to align already in the order given by a guide tree:

Seq A  GARFIELD THE LAST FAT CAT
Seq B  GARFIELD THE FAST CAT
Seq C  GARFIELD THE VERY FAST CAT
Seq D  THE FAT CAT
```

```
Step 1
Seq A  GARFIELD THE LAST FAT CAT
Seq B  GARFIELD THE FAST CAT

Step 2
Seq A  GARFIELD THE LAST FA-T CAT
Seq B  GARFIELD THE FAST CA-T
Seq C  GARFIELD THE VERY FAST CAT

Step 3
Seq A  GARFIELD THE LAST FA-T CAT
Seq B  GARFIELD THE FAST CA-T
Seq C  GARFIELD THE VERY FAST CAT
Seq D  -------- THE ---- FA-T CAT
```

Historically the most used of the progressive multiple alignment algorithms was CLUSTALW. Nowadays CLUSTALW is not one of the recommended algorithms anymore because there are other algorithms that create better alignments like Clustal Omega or MAFFT. MAFFT was one of the best contenders in a multiple alignment software comparison.

T-Coffee is another progressive algorithm. T-Coffee tries to solve the errors introduced by the progressive methods by taking into account the pair-wise alignments. First it creates a library of all the possible pair-wise alignments plus a multiple alignment using an algorithm similar to the CLUSTALW one. To this library we can add more alignments based on extra information like the protein structure or the protein domain composition. Then it creates a progressive alignment, but it takes into accounts all the alignments in the library that relate to the sequences aligned at that step to avoid errors. The T-Coffe algorithm follows the steps:

1. Create the pair-wise alignments
2. Calculate the similirity matrix
3. Create the guide tree
4. Build the multiple progressive alignment following the tree, but taking into account the information from the pair-wise alignments.

T-Coffee is usually better than CLUSTALW and performs well even with very different sequences, specially if we feed it more information, like: domains, structures or secondary structure. T-Coffee is slower than CLUSTALW and that is one of its main limitations, it can not work with more than few hundred sequences.

Iterative algorithms

These methods are similar to the progressive ones, but in each step the previous alignments are reevaluated. Some of the most popular iterative methods are: Muscle and MAFFT are two popular examples of these algorithms.

Hidden Markov models

The most advanced algorithms to date are based on Hidden Markov Models and they have improvements in the guide tree construction, like the sequence embedding, that reduce the computation time.

Clustal Omega is one of these algorithms and can create alignments as accurate of the T-Coffee, but with many thousands of sequences.

Alignment evaluation

Once we have created our Multiple Sequence Alignment we should check that the result is OK. We could open the multiple alignment in a viewer to assess the quality of the different regions of the aligment or we could automate this assesment. Usually not all the regions have an alignment of the same quality. The more conserved regions will be more easily aligned than the more variable ones.

It is quite usual to remove the regions that are not well aligned before doing any further analysis, like a phylogenetic reconstruction. We can remove those regions manually or we can use an especialized algorithm like trimAl.

Software for multiple alignments

There are different software packages that implement the described algorithms. These softwares include CLI and GUI programs as well as web services.

One package usually employed is MEGA. MEGA is a multiplatform software focused on phylogenetic analyses.

Jalview and STRAP a multiple alignment editor and viewer. Another old software, that has been abandoned by its developer is BioEdit.

In the EBI web server have some services to run several algorithms like: Clustal Omega, Kalign, MAFFT, and Muscle.

Phylogeny and evolution

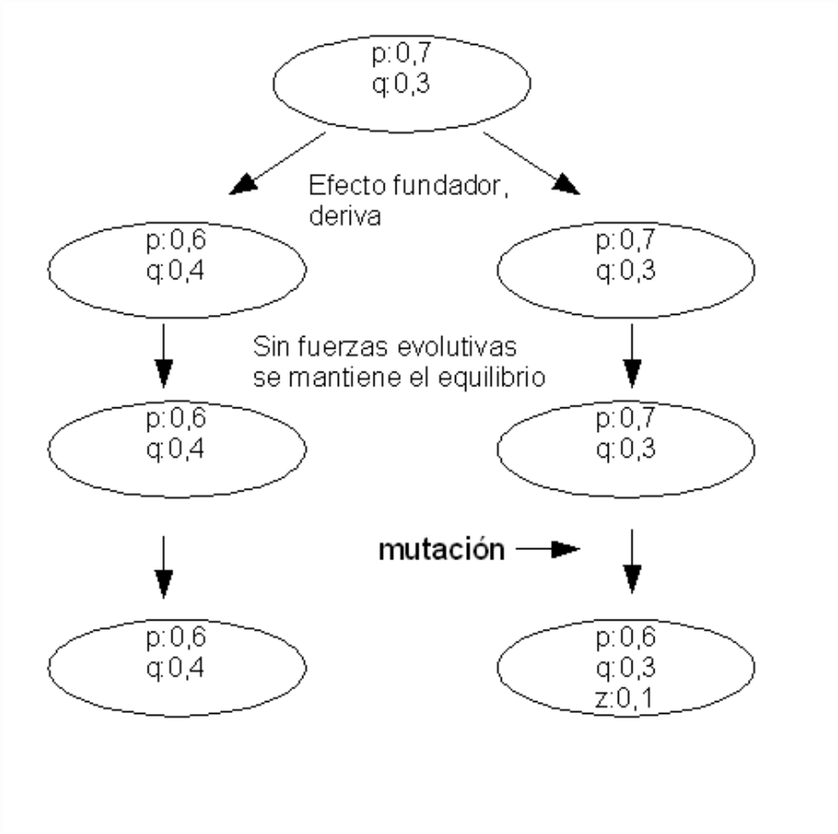
Speciation

Speciation is the evolutionary process by which reproductively isolated biological populations evolve to become distinct species. Two populations from the same species that are reproductively isolated, that do not have gene flow between them, can end up with time creating two new species incapable of having sexual reproduction between them.

Speciation mechanisms

A population is a group of individuals that live in the same greographical area and are capable of interbreeding mixing their genetic information.

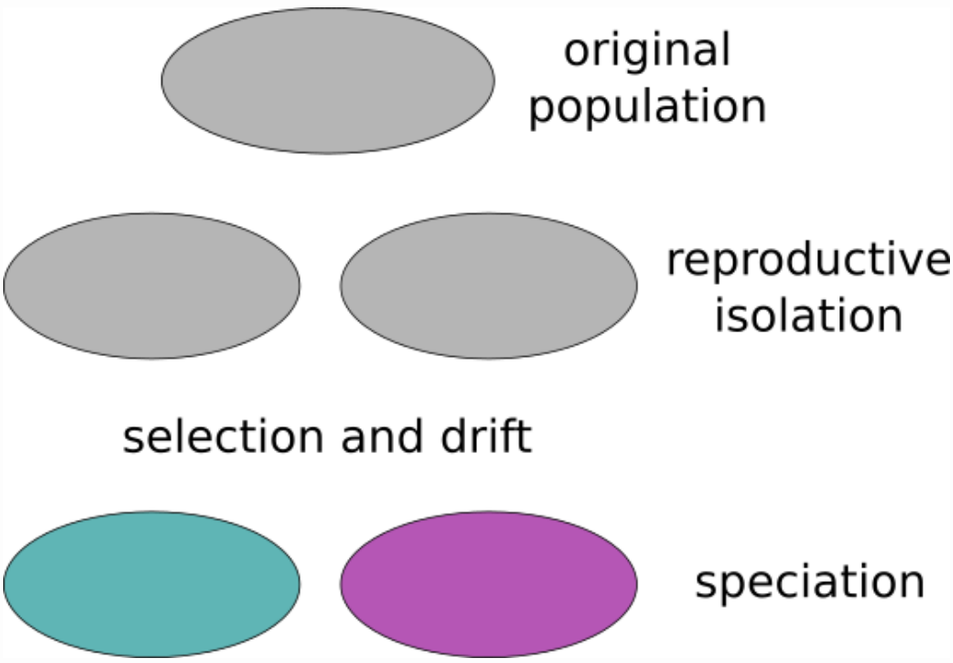
Individuals will interbreed more usually with other individuals from their same populations than from individuls from other populations. Selection, genetic drift, mutation and migration will affect in different ways to different populations and therefore different populations will have different genetic characteristics. Allelic frequencies will differ between populations.



A species is usually composed by different populations which have different characteristics. The species is maintained integrated by the genetic flow that goes from one population to another. If the genetic flow is low the populations will tend to differ with time. Different selective pressures acting in different populations will

also make them became different. If the isolation between the populations is maintained they will create races, subpopulations and finally species.

The isolation between different populations can be due to different causes. One common cause is the geographical isolation. The populations are split because of the geography (mountain ranges or rivers for example) or by distance (like South and North America). The speciation cause by this reason is called alopatric.



A sympatric speciation happens when the isolation between the species happens despite living in the same geographical region. These could be different reasons for a sympatric speciation like:

- Habitat or seasonal isolation. One species can reach the sexual maturation in a different species than other or might inhabit different ecological niches.
- Sexual or behavioral isolation. Both species have a sexual incompatible behavior.
- Mechanical isolation. The reproductive organs are not compatible any more.
- Postzigotic isolation. The zygote is formed but is not viable due to genetic or other causes. It might also be possible that the hybrid is viable, but it is sterile.

Microevolutive processes

Microevolution is the change in allele frequencies that occurs over time within a population.

The processes that underlie microevolution are: mutation, selection (natural and artificial), gene flow and genetic drift.

Mutation

Genetic variability is the pre-requisite for evolution. The other microevolutive processes will act upon the variability created by the mutation.

The mutations are:

- random and non-directional towards a goal.
- They create variation. Mutation is the only process that creates new variation.

Types:

- Point mutations, nucleotide substitutions.
- Small insertion and deletions
- Structural variants

Causes:

- Spontaneous
- replication errors
- DNA repair errors
- mutagens

Genetic drift

Genetic drift is the change in the allelic frequencies due to the random sampling of alleles to create a new generation. There wouldn't be any genetic drift in an infinite population.

You can simulate the random drift.

Try simulate what happens when you create smaller and larger populations. Simulate what happens with different starting allele frequencies.

Characteristics:

- It removes variability
- It is neutral
- Mechanism that controls most of the genetic variation

Selection

Selection is due to the differential reproductive success of different genotypes.

Selection removes variation created by mutation. It could be compared with a sculptor that removes fragments from a stone to create a statue.

Mutations can be with respect to selection:

- beneficial
- deletereous
- neutral

The fitness is the quantitative representation of the selection, it measures the contribution of an individual to the genetic pool of the next generation.

It has no sense to think on the fitness without taking into account the environment. One trait could be beneficial in one environment and deletereous in another.

Selection adapts the species to the environment and improves the fitness overtime.

Species

Variation within and between species

There could be genetic variation within and between species. When the genetic variation is dominated by the intraspecific variation it is advisable to do genetic population analyses and not phylogeny.

Phylogeny assumes that the variation within species compared with the variation between the species is negligible.

Species concept

A species can be defined as the group of individuals capable of breeding and have fertile offspring. This standard definition focuses on the genetic flow because the lack of genetic flow will make populations differentiate over time creating new species. But there are different problems with this definition:

- The capacity of breeding fertile offspring ranges from impossible to completely compatible in a continuos variation. Where should we trace the line that split the species.

- If two groups can potentially breed but they live in different continents do they belong to the same species?
- It does not account for the amount of morphological, physiological and ecological differences. What happens if we have two distinct groups that do not breed because of a geographical barrier and that are ecologically very distinct? Even if they could potentially breed are they different species?
- If the species has asexual reproduction, how does the gene flow definition applies?
- If the species had sexual reproduction but we only have fossils how can we determine if two individuals belong to the same species?
- If we study a species that has derived from an extant old species when shold we split the two species?

Trees and networks

Phylogenetic analyses assume that species evolve into new species and that there is no gene flow between those species once they have split. There can be some exceptions to this assumptions:

- Very close species can have gene flow between those because they can still produce fertile hybrids.
- There is horizontal transfer. Some genes can jump from one species to another without being transmitted by sexual interbreeding. This is specially the case in bacteria.

When there is gene flow between species the evolution is better represented by a network than by a tree.

These networks a typical of the populations that after they are split they can still have gene flow between them. In this case we could consider using population genetic analyses instead of phylogenetics.

A clear case of a network is the endosymbiosis of the mitocondria and the chloroplast. In this case the genes will have bifurcating trees, but the species evolution will be a network.

Introduction to phylogeny

Phylogenetic analyses try to infer the evolutive relationships between species. They try to build the correct topology, the order of splits in the ancestral species that created the extant species, and the genetic distances, that are related to the time passed since the splits.

The phylogenetic methods usually assume that the extant species analyzed were created by splits of the ancestral species in a bifurcating fashion. If this assumption is not met the result of the phylogenetic analysis might be misleading. For instance, if we are analyzing populations within a species there could be gene flow due to migration and that won't be reflected in the tree build by the phylogenetic analysis.

Phylogenetic analyses create the phylogenetic trees using the experimental evidences available. Some kinds of evidences are:

- Morphological data
- Genotypes
- DNA or protein sequences

Taxonomy vs phylogeny

Taxonomy is the science of defining groups, classifying, on the bases of similarity and shared characteristics. Phylogenetics is the study of the evolutionary history of the living beings. Both concepts are related, but they are not the same. In biology, Cladistics, we try to classify the species using their evolutionary history. We could create the biological groups based on characteristics not related with its history. We could classify according to morphological or ecological similarities not because of their history. For instance, herpetology is the study of amphibians and reptiles, despite the fact that cladistics would classify them in distinct groups. Some methodologies traditionally used in phylogenetics, like UPGMA trees, are also commonly used in taxonomy, even in non-biological taxonomies.

Nomenclature

Taxon and clade

A taxon is a group of organisms defined in a taxonomical analysis. The taxa can be families, genera, species, etc. Examples of taxa are: mammals, reptiles, insects or fishes.

A clade is a branch in a phylogenetic tree. It can be a group of species with their common ancestors.

All clades could be taxa, if somebody name them, but not all taxa can be clades. Only the monophyletic taxa are clades. For instance, the vertebrates are a taxon and a clade. But the taxa reptiles and fishes are not clades.

Monophyly, polyphyly and paraphyly

A monophyletic group is a taxon which forms a clade, meaning that it consists of an ancestral species and all its descendants.

A polyphyletic taxon is comprised by branches that do not originated from a common ancestor. For instance, worms would be a polyphyletic taxon.

In a paraphyletic taxon all their members originated from a common ancestor, but not all the descendants of that ancestor are included in the taxon. Reptiles or fishes are examples of paraphyletic taxa.

Trees, dendograms and cladogram

A phylogenetic tree is a representation of the inferred evolutive relationships, the phylogeny, of a group of clades. If you follow the diagram from one species in the tip to the ancestor species you will follow the evolutionary history of the species.

Phylogenetic trees depict two kinds of information, the topology, the pattern of the branching, and the length of the branches. The topology is related with the order in which the species split in the evolutionary history and the branches with the time or amount of change between the species.

A a phylogram is a phylogenetic tree in which the branch length should be taken into account.

By contrast, in a cladogram only the topology is relevant.

In an ultrametic tree of the branches from the common ancestor to the extant species have the same length.

If all genetic distances were proportional to the time since the split of the species all phylogenetic trees would be ultrametric, but this is seldom the case. This is known as the molecular clock hypothesis. Usually some branches evolve at a faster or slower pace. Some possible reasons for these changes are: selection or genetic drift. For instance, small populations will change faster due to drift and species in new ecological niches will suffer a stronger selection pressure and will change faster.

Equivalent topologies

We can create alternative, but equivalent, graphical representations of a tree. We have to be cautious when judging which trees have different topologies because we can have different representation of the same underlying tree.

```
|          --------- A
|    ------|
|    |     --------- B
|    |
|    |       --------C
|    --------|
|            --------D
|
|          --------- B
|    ------|
|    |     --------- A
|    |
|    |       --------C
|    --------|
|            --------D
|
|          --------- B
|    ------|
|    |     --------- A
|    |
|    |       --------D
|    --------|
|            --------C
```

Rooted and unrooted trees

A phylogenetic tree can be represented with or without a root. In a rooted tree the there is a node that corresponds to the common ancestor of all the leaves of the tree. In this case all nodes represent the most recent common ancestor of the clade that derived from that ancestor.

```
|        --------- A
| ------|
| |     --------- B
| |
| -----------------C


|          --------- A
|    ------|
|    |     --------- B
| --|
|    |
|    -----------------C
|
|          --------- A
|    ------|
|    |     --------- B
|    |
|    |
|    ----------------C
```

Phylogenetic reconstruction methods do not create rooted trees. We have to determine which is the ancestor node. We could do it by assuming the molecular clock hypothesis In this case the most distant node of the extant nodes will be the most ancient one. The problem with this approach is that we can not be sure that the molecular clock hypothesis is true for all phylogenies.

So the most common way of creating rooted trees is to include some taxon in the phylogenetic reconstruction that we are sure that is the most distant and unrelated one. For instance, if we would root a tree of the mammals we could use a crocodile.

Polytomy

A polytomy is a unresolved node in which several branches appear.

The phylogenetic reconstruction methods assume that all splits are in two, so all polytomies would be due to lacking phylogenetic signal or to lacking reconstruction methods. To solve a polytomy we need evidence that correspond to mutations that appeared in the period in which the species related to the polytomy split.

Phylogenetic inference based on sequences

Phylogenies can be inferred using different kinds of evidence like:

- Morphology
- Molecular markers
- Presence and absence of genes.
- DNA or protein sequences

But the most common approach is to use sequences when they are available.

In any case we need characters that are similar because they have a common ancestor, not the characters that are similar because they were selected to adapt the organisms to a similar ecosystem niche. For instance, if we consider dolphins and sharks to be closely related because they share a similar shape we would be mistaken. The problem lies in the character chosen. We evaluate the phylogenetic relationships taking into account the similarities in some characters. If those similarities are due to a common ancestor, for instance we have for limbs like the cats because we have a common ancestors. Those characters that are similar because they originated from a common ancestor are called homologous characters. The characters that are similar, but not because they have a common ancestor are said to be analogous.

For morphological characters can be difficult to know if a character is similar between two species because it is homologous or analogous because selection can create analogous structures in organism that face the same ecological problems. For the sequences this is seldom the case. Two sequences that are similar are similar in most of the cases because they are homologous. The molecular function is not so directly tied with the sequence. Different sequences can have the same function, so it is unlikely that selection creates molecules with the same sequence, even if it creates sequences with the same function.

When we are using sequences to build a species tree we assume:

- Each sequence is correct and it belongs to the organism that we are studying.
- Sequences are homologous, they evolved from a common ancestor.
- Each position in the multiple sequence alignment is homologous in all sequences.

- All sequences correspond to extant species and no extant species originated another extant species. All extant species will be leaves in the tree.
- Mutations happened at random
- Different positions evolved independently.
- There is no genetic flow between the different species after their split. Evolutionary history is a tree, not a network.

The sequences used can have enough phylogentic signal to infer the phylogeny in every detail, but that can be not true. We have to check using some statistical method which features of the phylogeny are statistically significant and which are not.

Usually when we interpret a species phylogeny we assume that the sequence variation within species is very small compared with the variation between species. This won't be true for populations.

Uses

Phylogenies can be used to study the species evolution or the evolution of genes.

One common use case is to use sequences of extant species to infer their evolutionary history.

Evolutionary history can be studied for the broad taxonomic ranks or within species. We can be interested in build a tree for all metazooans of just for the HIV viruses.

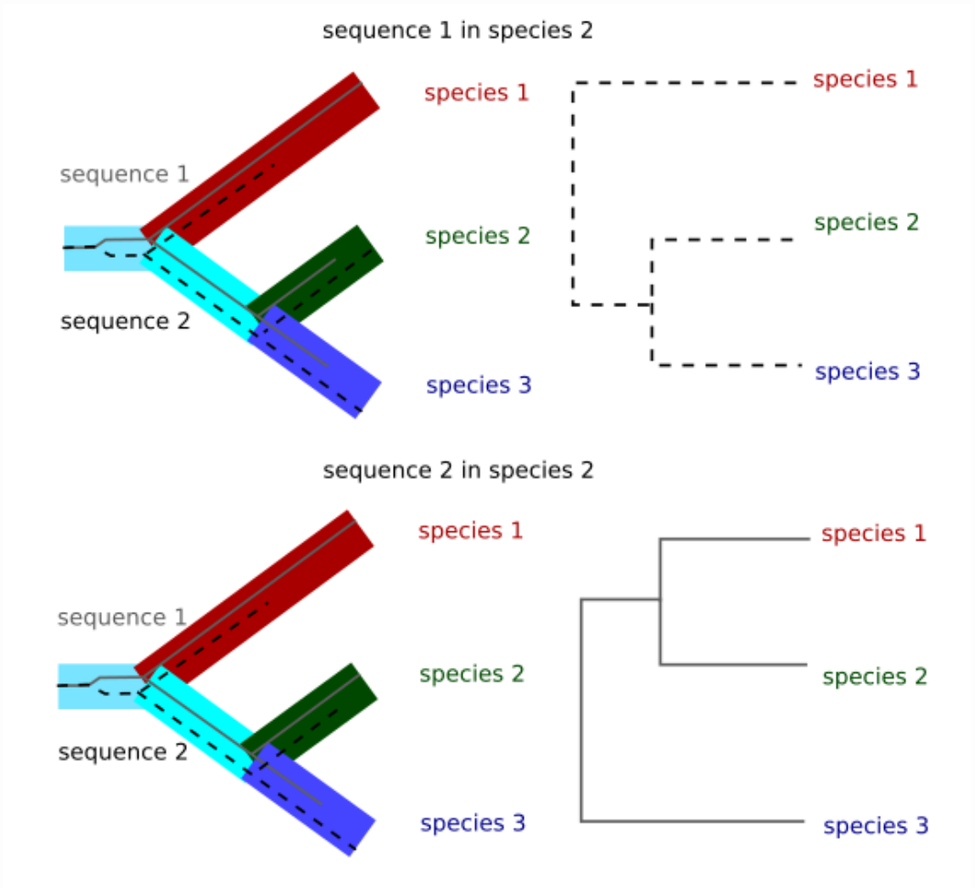
We can also use phylogenies to study the evolution of genes and their functions in different organims.

Species tree vs gene tree

The phylogenetic history of the species and the genes of those species should be the same in general, but there might be differences. Sometimes we can find genes that have a different history than the species that host them.

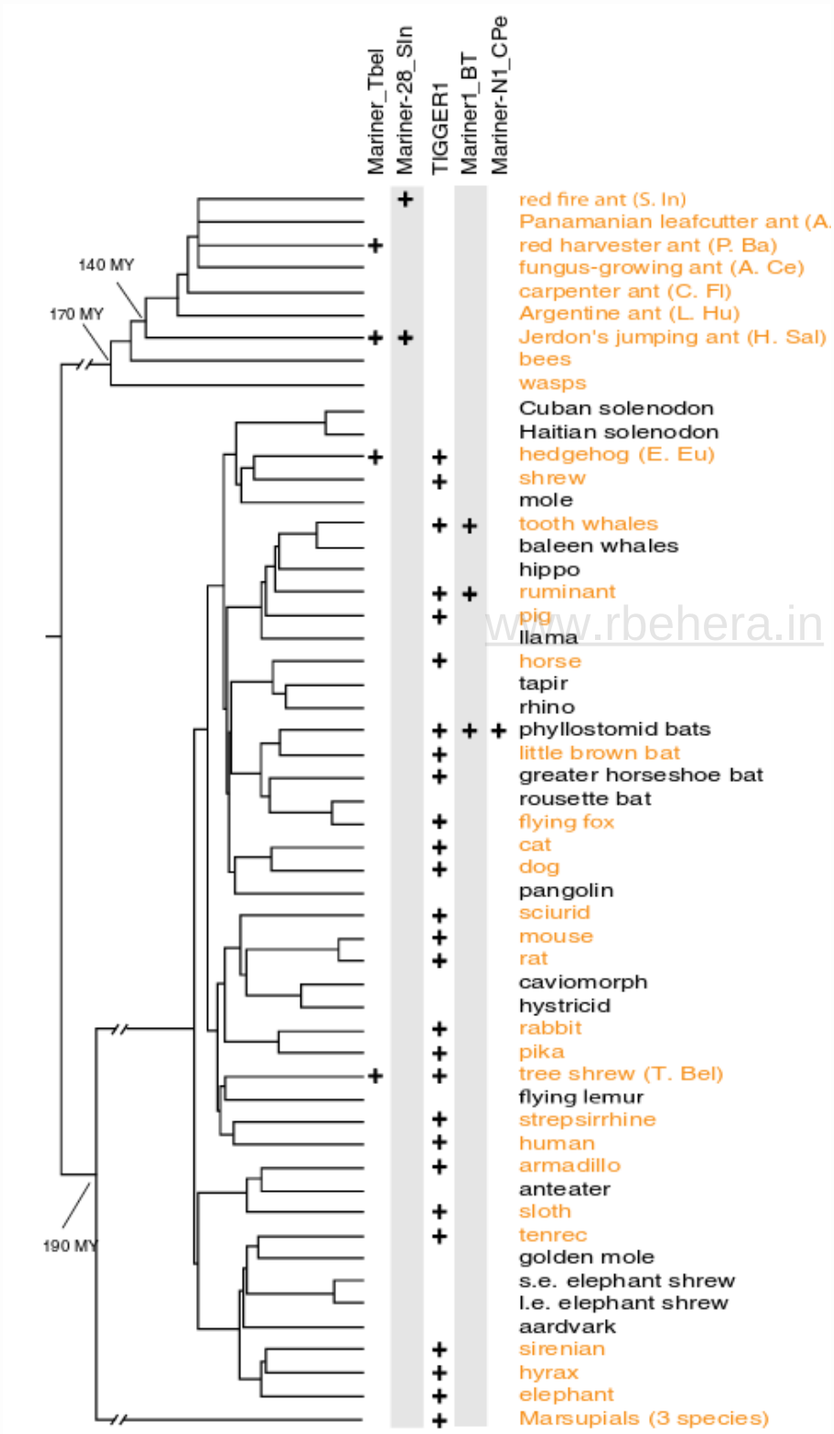
When we are building a species tree we assume that the sequences used are representative of the species evolution.

When we are studying very close species it is common to have incomplete lineage sorting.



Imagine that you have two alleles for one gene in one ancestral species. Those alleles will have differences between them. If we sequence only one allele from the extant species and this allele is chosen at random in those species the final tree will depend on the alleles chosen for each species. This case is only relevant in very close species because if the species are different enough the number of mutations since the split from the original species will be much larger than the differences between the two original alleles and those differences will be irrelevant for the phylogenetic reconstruction. This case is, for instance, relevant in the tree of chimpanzee, gorilla and human. Depending on the region of the genome chosen the gorilla might appear to be closer to human than the chimpanzee.

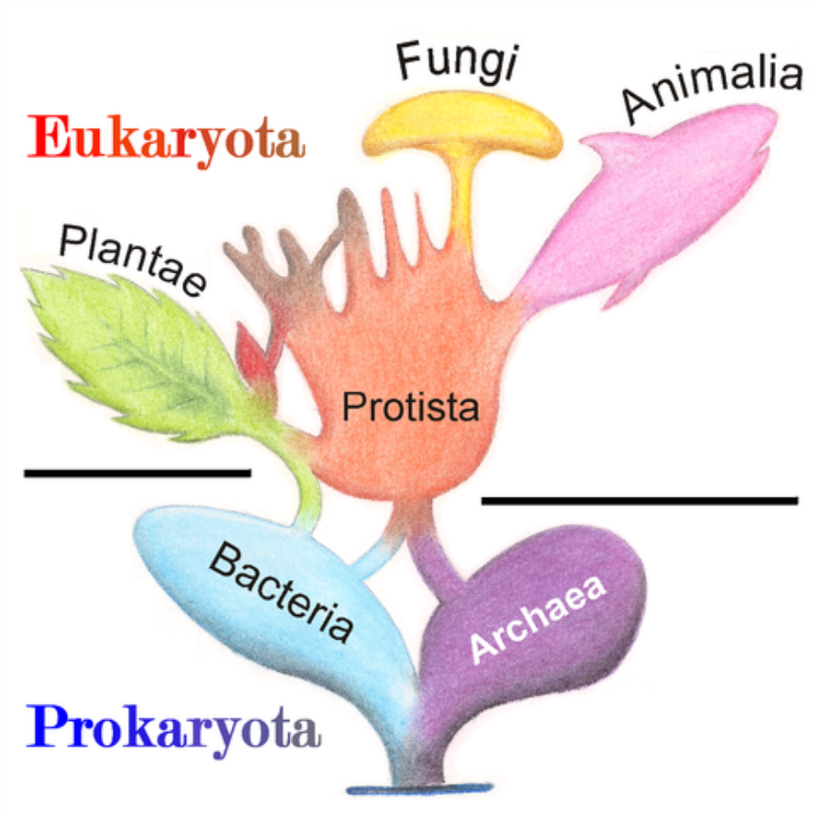
Another common phenomenon that differentiates the species and the gene tree is horizontal transfer. Usually genes move from parents to offspring, but in some cases they can move horizontally from one species to another. This is common, for instance, within transposable elements. One example is the case of the mariner element. The transposable element mariner has been transfered between insect and mammal species and there are insect and mammalian sequence with identities higher than 95%.



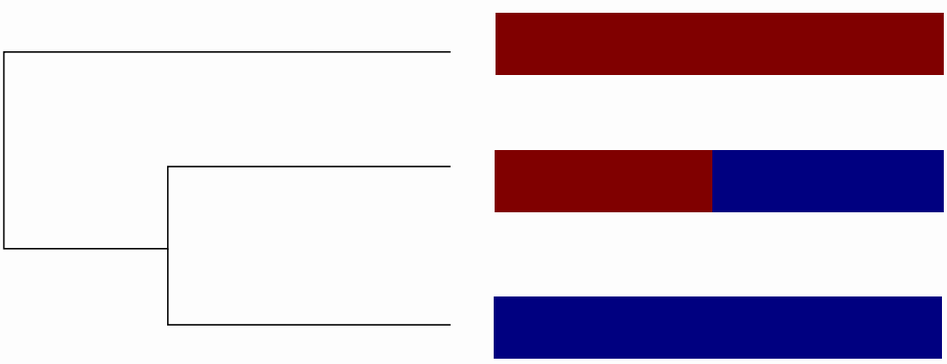
In the figure it is show the location of different subfamilies of the mariner element: Mariner_TbeI, Mariner-28_sln, Mariner1_BT and Mariner-N1-CPe. Mariner_TbeI is found in the Tree shrew and in a couple of ant species.

Horizontal gene transfer is also very common in prokaryotes.

Another case in which there are discrepancies between the species tree and the gene trees is when species interchange genetic material. This can happen when species are very closely related and they still produce fertile offspring. Another case happened when the eukariotes evolved from a fusion of an archea and a bacteria.



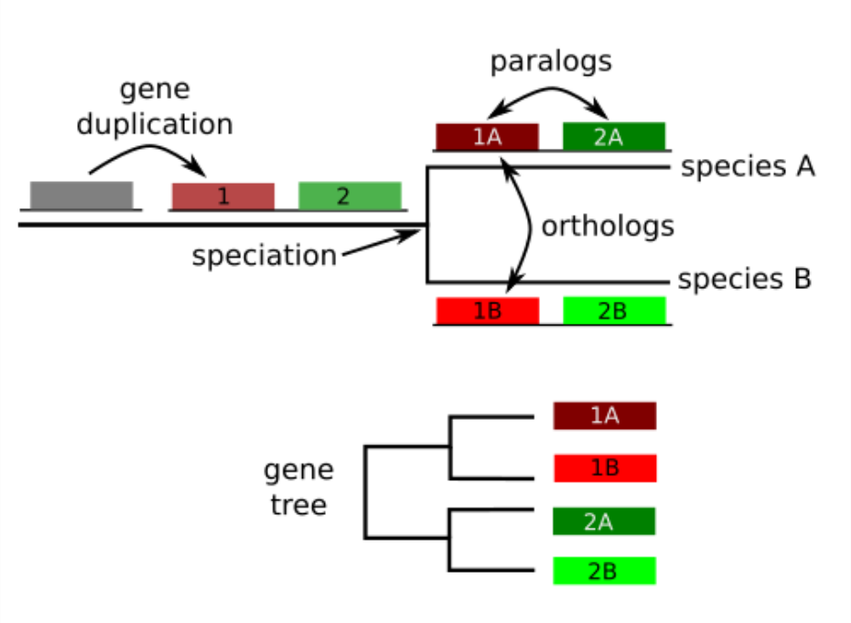
You also have to take into account that when there is genetic interchange between species recombination might happen and you can end up having sequences that have stretches that have had different evolutionary histories. Phylogenetic methods assume that there is no recombination.



Nowadays it is common to build thousands of gene trees from many genes of the genome and infer the species tree from those trees. This is the area of phylogenomics.

Gene families

A gene family is a set of similar genes created by ancestral duplications.



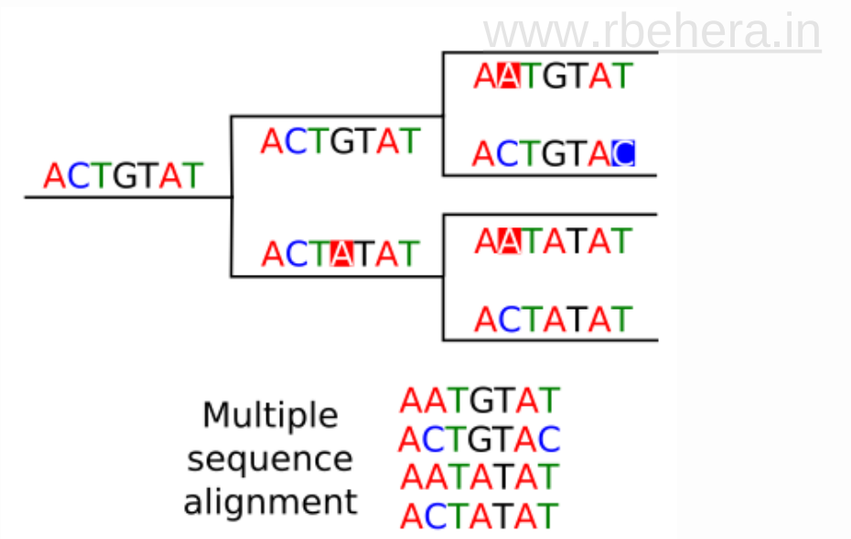
All genes that originated from the ancestral copy are homologous, but we can further classify them:

- Homologous sequences are orthologous if they are inferred to be descended from the same ancestral sequence separated by a speciation event
- Homologous sequences are paralogous if they were created by a duplication event within the genome.
- Homologs resulting from horizontal gene transfer between two organisms are termed xenologs.

Multiple alignment as evidence for phylogenetic inference

Phylogenetic trees are usually build from multiple sequence alignments.

We asume that aligned positions for each sequence correspond to homologous positions and the the differences are due to mutation that occurred along the evolutionary history.



The higher the quality of the multiple sequence alignment the better will be our phylogenetic reconstruction. If we suspect that there are misaligned regions it is better to remove them before doing the phylogenetic analysis. We can check manually the multiple sequence alignment to remove suspicious regions. In general the regions that accumulate more mutations will be more difficult to align and more prone to misalignments.

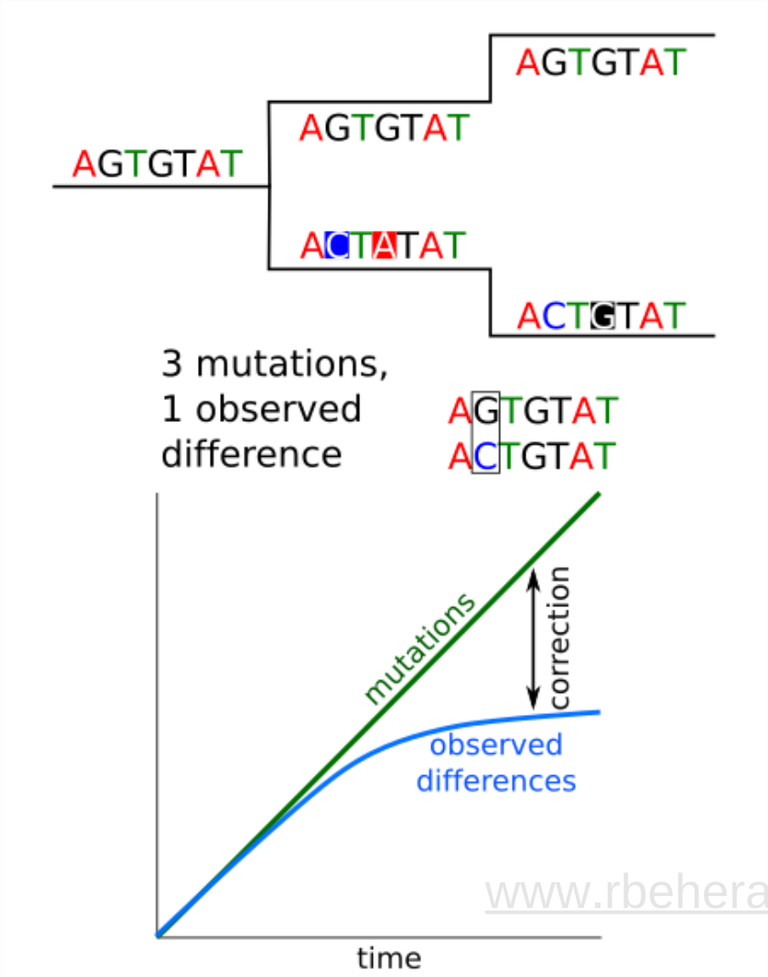
This pruning of misaligned regions can also be done automatically with specialized software like Gblocks or TrimAl. These programs remove regions according to its level of conservation, number of gaps, etc.

Models of nucleotide substitutions

Sequences accumulate mutations with time, so differences between homologous sequences inform us about the evolutionary distance and the time since those sequences begun their split. The more different two sequences are, the more time should have passed since their split, but there are several confounding factors for this simple assumption.

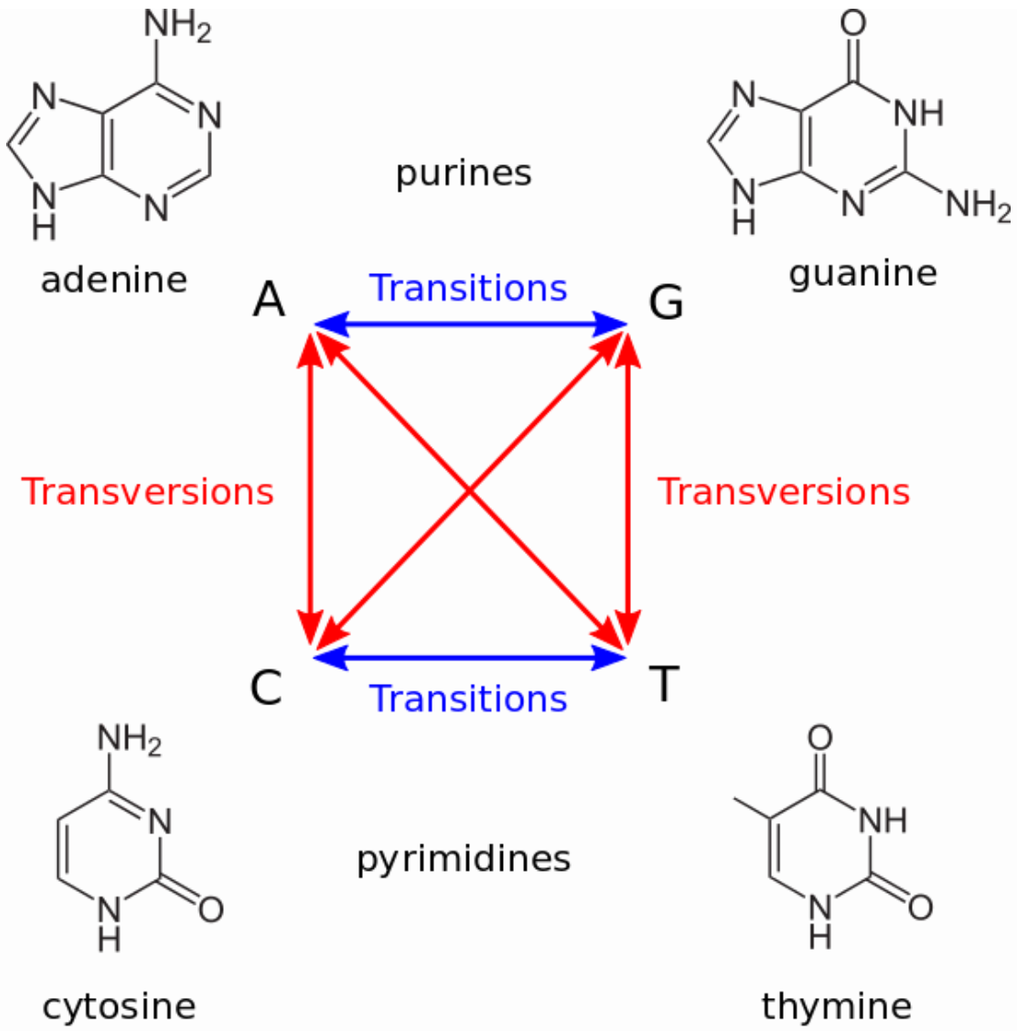
Mutations vs observed changes

We could think that counting the number of differences between two sequences we are counting the number of mutations between them, but that is not the case. The several mutations can occurred at the same position and we would count just one difference or maybe even none if the mutation reverted the sequence to the original sequence. This problem has to be corrected if we want to account for the real number of mutations.



Transitions vs transversions

A transition is a change of one purine nucleotide by another one: A to G or G to A or a pirimidine nucleotide by another one: C to G or G to C. So a transition is a change of one nucleotide by another chemically similar. A transversion is a change of purine by a pirimidine or viceversa.



Transitions and transversions do not occur with the same frequency, transitions are more likely. Substituting a ring structure for another single ring structure of the same type is more likely than mutations between different rings. Also, transitions are less likely to result in amino acid substitutions (due to wobble base pair), and are therefore more likely to persist as “silent substitutions” So, if we are interested in accounting for the time since the split of one species it would be better to count transversions ans transitions independently because they accumulate at a different rate.

Models of nucleotide substitution

The models of nucleotide substitution account for the process in which one sequence is changed into another. These models account for the relative frequencies of the different possible changes. They correct for as many confounding factors as possible to account for the true time since the split of the species.

These models usually assume that different positions in the sequence alignment evolved independently. This is true for sites evolving neutrally and it could be not true for some selective pressures.

These models differ in the assumptions that they made:

- All mutations are equally probable or not.
- All sites evolve at the same rate or not.
- All nucleotides are found at the same frequency or not.

Popular substitutions models

The Jukes and Cantor model is the simplest substitution model. It assumes that all mutations are equally probable, that all nucleotides are found at the same frequency and that all sites evolve at the same rate. In this model there is only one parameter, the substitution rate at which the mutations occur.

The kimura model distinguishes between the substitution rate for transitions and transversions. It assumes that all bases are found at the same frequency and that all sites evolve at the same rate.

There is a generalised time reversible model that allows for different mutation rates between all nucleotides and different frequencies for the nucleotides.

There are also models that assume different mutation rates for different positions in the alignment. This account for sites that are more conserved than others due to selection.

Choosing between models

The best model for our phylogeny depends on the sequences that we are using.

The evolution of different sequences can be best modeled by one substitution model than other. For instance, if there are sites strongly selected and sites that are neutral it might be better to use a model that allows for different mutation rates across positions.

There is also another factor to take into account. The phylogenetic signal contained in the sequence alignment that we are using is limited and the more parameters a model has the more signal we need if we don't want to overfit the model. An overfitted model would describe our data by adjusting the noise in it as if it was the reality.

Thus, the model to use would be the model that best fit our data, but taking into account the amount of phylogenetic signal to avoid overfitting. So the model will depend on the sequence alignment and it has been shown that the model choice might influence the result of the phylogeny.

There are different programs to calculate which is the model for our data. They create a rough first tree and from that they try all the models and check how the fit the data. One of such programs jmodeltest.

Methods of phylogenetic reconstruction

We can divide the methods in:

- heuristic methods based on distances
- Maximum parsimony methods
- Maximum likelihood
- Bayesian

Phylogenetic reconstruction based on distances

Genetic distance

The genetic distance is a measure of the degree of difference between to sequences.

There are different statistical measures to calculate the distance between two sequences. In theory we could just use the number of differences between sequences divided by the length of the alignment, but as we have seen we have to account for the number of mutations not for the number of differences. Many sites will have had several mutations and by counting the number of differences we are underestimating the genetic distance. So, we have to use a method to estimate the genetic distance that uses a nucleotide substitution model.

If we have several sequences aligned we can calculate the distances between any pair of them. These will be the pairwise distances and with them we can calculate a matrix.

	Human	Chimp	rat	Mouse
Human	0	0,9	0,5	0,48
Chimp		0	0,51	0,49
rat			0	0,85
Mouse				0

Tree generation from distance matrices

There are several methods to generate trees from a pairwise distance matrix. They are general statistical methods used in different fields, not just in phylogenetics. They might be used for any problem that involve creating hierarchical classifications. The most common of these methods in phylogenetics are UPGMA and Neighbor-joining.

These heuristic methods are very fast. They can be used with huge distance matrices and they do not depend directly on the sequence length because all they take is the pairwise genetic distance and they do not consider the alignment per se. They run very fast without much memory.

We can generate a tree from any distance matrix, but not all distance matrices are equally well described by a tree. Some matrices, for instance, might be better described by networks than by trees.

Once we have generated a tree it is advisable to check how well the tree matches the original distance matrix. One way of doing that is to calculate a cophenetic correlation index. To do it we calculate a new distance matrix from the tree and we calculate the original distance matrix with the new matrix generated from the tree. A high correlation would indicate that the tree is a good representation of the original matrix.

UPGMA

UPGMA is a clustering method based on looking for the most similar pairs. One the most similar pair is found the distance matrix is recalculated with this pair as an entity.

This method will generate ultrametric trees so it is advisable to use it only if we are sure that the molecular clock hypothesis is a good match for our data.

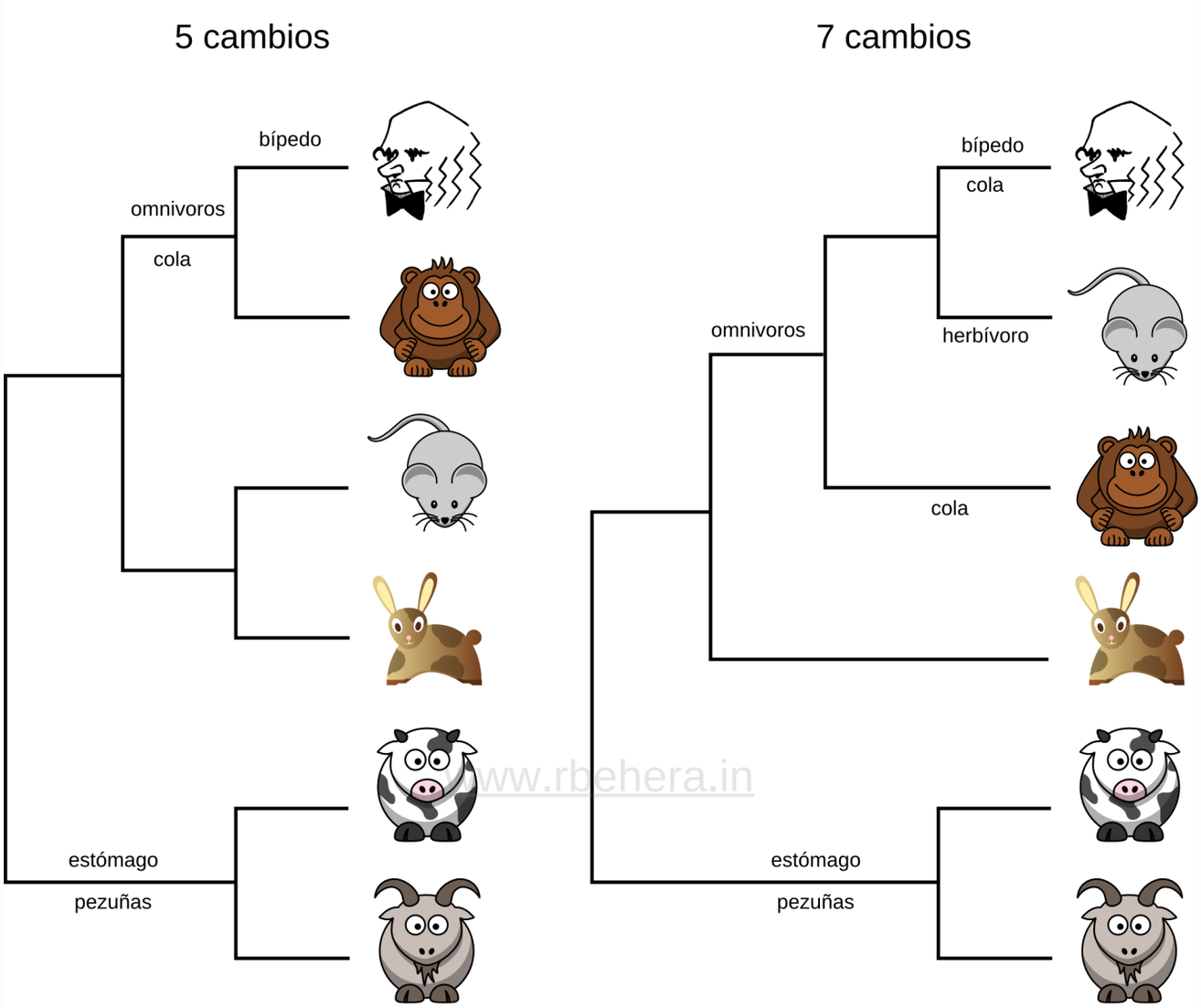
Neighbor-joining

Neighbor joining is very commonly used because is fast and it has no restriction regarding the molecular clock. It will generate non ultrametic trees with branches that span different lengths.

Maximum parsimony

The maximum parsimony approach tries to obtain the tree that requires the least number of changes to explain the character matrix given, e.g. the multiple sequence alignment. The idea behind it is that the simplest explanation should be the correct one.

	Alimentación	Estómago	Pezuñas	Bípedo	Cola
Hombre	Omnívoro	Simple	No	Sí	No
Chimpancé	Omnívoro	Simple	No	No	No
Ratón	Herbivoro	Simple	No	No	Sí
Conejo	Herbivoro	Simple	No	No	Sí
Vaca	Herbivoro	Compuesto	Sí	No	Sí
Cabra	Herbivoro	Compuesto	Sí	No	Sí



To chose the most parsimonious tree the method should, in theory, evaluate all possible trees. For each tree should calculate how many mutations need to account for the given character matrix. After having that information it should chose the trees with the least number of mutation. It could be one tree or several that have the same number of mutations.

In practice it is not possible to evaluate all trees because the number of trees grow very fast with the number of taxa. Only with few taxa would it be possible to check them all. So the parsimony methods use a heuristic to chose the most likely trees and to evaluate the number of mutations only on those.

For the purpose of a maximum parsimony analysis not all characters are informative.

```
A   aat tcg ctt cta gga atc tgc cta atc ctg
B   ... ..a ..g ... .t. ... ... t.. ... ..a
C   ... ..a ..c ... ... ..t ... ... ... t.a
D   ... ..a ..a ... ..g ..t ... t.t ..t t..
    1     2   3           4               5
```

Position 1 is not informative for parsimony because it is invariant and it would add zero mutations to any possible tree. So it won't differentiate between trees.

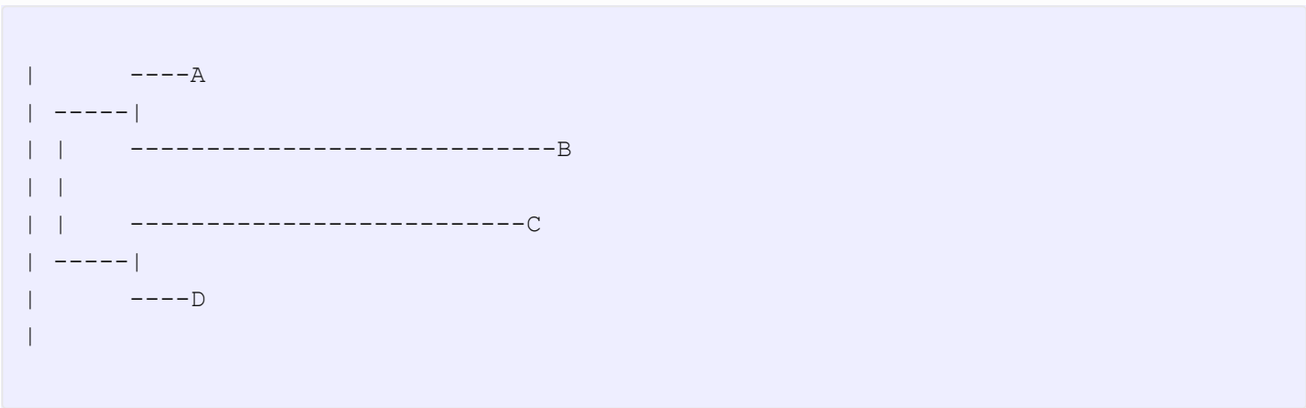
Position 2 does change, but only in one sequence. It is not informative for parsimony because it would add one mutation for any possible tree. These characters are called autapomorhies.

Position 3 is to variable because it changes in all individuals and it will contribute with 3 mutations to any tree.

Other methods do use these non-informative sites for parsimony and they have an influence in the genetic distances that they calculate.

Positions 4 and 5 are informative for parsimony because they are shared by some species and not by others and they would contribute with different numbers of mutations to different trees. These positions are called synaptomorphies.

The maximum parsimony method was very popular in the past, but nowadays it seldom used because it has been shown that it has some statistical problems. There are cases in which this method won't give us the correct topology. Moreover, the more phylogenetic data that we have the worse it will behave in these cases. These problems arise, for instance, when we have a tree in which some taxa have evolved faster than other and the problem is called long branch attraction.



If we try to reconstruct this phylogeny using maximum parsimony it will create a tree in which the taxa B and C are together in the base of the tree.

Maximum likelihood

Maximum likelihood is not used just for phylogeny, it is a very useful concept with wide application.

likelihood

In common language probability and likelihood are used as synonymous, but they are not in technical language.

Imagine that we are throwing a coin 10 times. If we assume that there is a probability *p* of landing heads we can calculate the probability of observing a particular outcome, like having 5 heads and 5 tails or 6 heads and 4 tails and so on. So we have a process with some observed outcomes, than we can name as *O* and some parameters that influence those outcomes that we can call *M* (M for model). Thus we can calculate the probability of the outcome given the parameters is P(*O*|*M*).

In real life we usually don't know the values for the parameters that define *M*. For instance, in the case of the coin we do not know the value for the probability of landing heads *p*. All we know is that we can do some observations, throwing the coin, and obtain some outcomes *O*. So we have to estimate *M* from our observations *O*. A natural way of estimating *M* is to find the parameters that maximize the probability of having observed *O*. So, we can define a function that has *O* as a given and has the parameters of *M* as variables. This is called the likelihood function, or just likelihood: L.

We can maximize L and in that way we calculate the parameters of the model that maximize the probability of having observed our data.

Example with a coin

We have a coin with a probability *p* of landing heads and 1 - *p* of landing tails. For a perfect coin *p* should be 0.5, but we want to check if our coin is perfect and we want to infer *p* from the observations that we have done.

We throw the coin n times and we get x heads and (n - x) tails. We want to calculate *p* from those observations.

The probability of having observed x heads and (n - x) tails is related to *p* by the following function:

P(x p, n) = n! / (x! * (n-k)!) * p^x * (1 -p) ^(n - x)

We can maximize this function for the variable *p* and thus we can calculate which value of *p* gives the maximum likelihood of observing x. To do it we can derive the function and ask for the derivative to be zero and to the second derivative to be negative. If we do just that we get:

p = x/n

You can also read the full demonstration.

To this estimation of the value of p we call it maximum likelihood estimation.

Maximum likelihood and phylogeny

We can use the maximum likelihood approach to look for the most likelihood phylogenetic tree. This would be the tree that makes the data that we have observed more probable.

To do it we need some observation, the multiple alignment. We also have to chose beforehand the some mutation model that we want to assume.

For each possible tree we will calculate the probability of the data being generated by the different trees and we will chose the tree that makes the data most probable. For each tree it will also calculate the parameters of the model that makes the data most probable.

It is not possible to inspect every possible tree because the number of trees grows very fast with the number of taxa so these programs use heuristics to inspect only the most likely trees.

This method uses the phylogenetic information present in our data in a more efficient way than the distance based methods and the maximum parsimony method. So given an alignment it might generate a better tree than the other ones.

The main problem of the method compared with the distance based methods is that it is computationally more costly. We can use it now for moderately big alignments because the computers are now quite powerful.

Bayes

Bayesian statistics based in Bayes theorem. The theorem allows us to calculate conditional probabilities, the probability of an event A given that other event B has happened.

```
p(A|B) = p(A) * p(B|A) / p(B)
```

This theorem is the base of the bayesian inference. We calculate the probability of an hypothesis or model (*M*) given some observations (*O*).

```
p(model|observations) = p(model) * p(observations|model) / p(observations)

p(M|O) = p(M) * p(O|M) / p(O)
```

p(observations|model) is the probability that we used in the maximum likelihood approach. It was the probability of having those observations given the model and the parameters that we had assumed. The probability that we calculate in the bayesian approach is the probability of the model given the observations.

The interpretations of this probability is more straightforward, it is just the probability of the model or hypothesis given the data that we have observed.

Example with a coin toss

In the movie The Dark Knight Harvey Dent before becoming Batman's enemy Two-Face picks between different paths by tossing a coin. It tosses the coin several times in the movie and it always the result is heads, never tails. The question is: when are we allow to suspect that there is something funny with Dent's coin?

We can think of two hypotheses: the coin is fair and has a head and a tail (H&T) or the coin just has just two heads (2H).

After Dent has tossed the coin for the first time we have 1 observation, one head, and we can calculate the probability of the coin having a head an a tail or just two heads using Bayes' theorem.

```
P(2H|1 observation) = p(2H) * p(1 observation|2H) / p(1 observation)
P(H&T|1 observation) = p(H&T) * p(1 observation|H&T) / p(1 observation)
```

The probabilities of having observed 1 head in 1 toss are easy to calculate for both models:

```
   p(1 observation | 2H) = 1
   p(1 observation | H&T) = 0.5
```

p(2H) and p(H&T) are the probabilities of the models without taking into account the observation, the probabilities of the models prior to the observation, and they are called prior probabilities. These probabilities can not be calculated from the data available. We have to assume values for the prior probabilities that look reasonable to us. For instance, in this case we could assume that having a coin with two faces is very weird. Alternatively we could assume that since we are watching a movie based on a comic in which a character named two-head appears we might assume that p(2H) is quite high. We could also assume different prior probabilities and we could check what happens in any case.


```
    Case 1: p(2C) = 0.0000000001
    Case 2: p(2C) = 0.5
```

Finally, we have to calculate the probability ob ser observation independently of the models. That means, the probability of the observation under any model considered being true.

```
p(1 observation) = p(2H) * p(1observation|2H) + p(H&T) * p(1observation|H&T)
```

Given the prior probabilities we can calculate everything

```
p(2H|1observation) = p(2H) * p(1observation|2H) /(p(2H) * p(1observation|2H) + p(H&T

Case 1 (p(2H)=1e-10): P(2H|1observation) = 2e-10
Case 2 (p(2H)=0.5): P(2H|1observation) = 0.6666666666666666
```

In both cases we have updated the probability of both hypotheses, but both posterior probabilities are very different. Posterior probabilities depend on our observation and on the prior probabilities.

What would happen if Havery Dent would continue tossing the coin and getting heads all the time? In that case the posterior probabilities would converge on very close posterior probabilities no matter what prior probabilities we start with. So, with enough observations posterior probabilities can be independent on the prior probabilities.

Epistemological implications of bayesian inference

We can think about bayesian inference as a way of learning. Every time we do an observation we update our prior knowledge (pior probabilities about the models and hypotheses) with the new evidence/observations

and we get new knowledge (new posterior probabilites).

```
knowledge = prior knowledge + new evidence
```

Different evidences should update your knowledge in different degrees. Imagine that somebody tells you that is capable of curing your influenza infection by treating you with a magical sleight of hand. So he does the trick and you get cured after a couple of days. Would you think that the infection was cured by his magic? No, because you would had been cured anyway by your own means. Most influenza infections are cured even if no treatment is given. It would be a very different kind of evidence if you could treat 100 people with the magic sleight of hand and cured them in a mean time of one day while other 100 people untreated are cured in a mean time of 3 days. That would be a much stronger evidence because the treatment is doing something that we do not see in the people that was not treated.

This is reflected in bayesian inference in the power of the evidence. The power of an evidence depends on the probability of having observed the evidence if the hypothesis is true *and* on the probability of having observed the evidence even if the hypothesis is *not* true.

```
p(O|M) / p(O)
```

Imagine that we want to detect the expression of a gene and we design a pair of primers located in the first exon of a gene. We do an RNA extraction, we retrotranscribe and we do the PCR. We get the in an agarose gel the band that we were expecting. Could we conclude that the gene is being expressed? No. Why? Because that band could be due to a contamination of genomic DNA of the RNA. So the band would appear even if the gene is not expressed. To be sure that the gene is expressed we have to set proper controls in the experiment. For instance, we could carry out two experiment one treating with DNAase the RNA before doing the retrotranscription and another treating with RNAase. If in the RNAase case the band disappears and in the DNAase case the band is still there we might conclude that the expression is real. This evidence is difficult to explain if the gene is not expressed. To avoid these problems the primers used to detect gene expression are usually designed in different exons to use the intron in the middle to avoid the expression due to genomic DNA contamination.

Every time we do a PCR reaction we include a negative control to be sure that our band is not due to a contamination but to our samples. It is also a good idea to include a positive control to be able to interpret the case in which we get no band. Has the PCR reaction failed?

A doctor does an inmonulogical tuberculosis detection test to you. The result is positive. Are you infected with tuberculosis? Probably not. If the analysis is well done they have detected antibodies in you. That suggests that you have had contact with the bacteria at some point, but not necessary that you are infected now. This test would be just a preliminar evidence, but to reach a solid conclusion we would have to do other analyses.

Prior probabilities affect our conclusions and should be taken into account.

You participate in a program to evaluate the prevalence of HIV in the standard population. The doctor informs you that the test for HIV presence was positive in your case. Are you infected? The doctor explains to you that the false positive rate of the test is 5%. That means that for every 100 analysis in non-infected people 5 turn out to be positive. Which is the probability that you are really infected? No, it is not 95% and you can not calculate that probability unless you know the prevalence of the HIV infection in your population (the prevalence). This is the base rate. Let's assume that in your population 2 out of 100 people are infected. Imagine that we do 1000 analyses to 1000 different people. How many analyses will be positive and how many negative?

```
True positives = 1000 * 2 / 100 = 20 people
False positives = 1000 * (100 - 2) / 100 * 0.05 = 49 people
```

If we do not have any other evidence and we take into account the base rate (as we should) we have to conclude that even after being a positive in the HIV detection analysis it is easier not to be infected than to be infected. This is know as the false positive paradox and it has implications in very different fields like health or antiterrorist prevention. This is one of the motive why there are no widespread campaigns to detect

some medical conditions in the population at large. The health, physiological and monetary costs of the false positives should be taken into account. We have a tendency of forgetting about the base rates and prior probabilities. This is a logical fallacy named as base rate fallacy.

Extraordinary claims

If in our research we reach an extraordinary conclusion, one far fetch result given the prior knowledge we have to provide also extraordinary evidences to back it up.

“Extraordinary claims require extraordinary evidence.” Carl Sagan

The case of Barry Marshall provides an example of an extraordinary claim back up by extraordinary evidences. He is a doctor that infected himself with *Helicobacter pylori* to show that, despite previous knowledge, this bacterium could case the peptic ulcer. If *H. pylori* was not capable of causing the ulcer he would not get peptic ulcer, but he did develop peptic ulcer one week after being infected. After the onset of the disease we was treated with antibiotics capable of killing *H. pylori*. After taking them he was cured. This evidence did also baked up the claim that *H. pylori* could cause a peptic ulcer despite the previous knowledge. After that first test clinical tests were set up to check if the peptic ulcer could be cured with antibiotics and they also backed up the hypothesis. So the conclusion was clear: *H. pylori* was capable of causing peptic ulcers. Marshall proposed an extraordinary hypothesis and provided extraordinary evidence to back it up and he was given the Nobel price for its contribution to medicine.

Prior probabilities criticisms

The main criticism to bayesian inference is directed against the evaluation of the prior probabilities. This evaluation is somewhat subjective. Two researchers can propose different prior probabilities for the same hypotheses because the judge different prior evidence in a different way. But trying to ignore the problem by not using prior probabilities in an explicit way just sweeps the problem under the carpet. The alternative of not using the previous knowledge is not to use it, to start always to the start of the research. We have to be aware of the dependence of the scientific inquiry on our previous knowledge. We have to make an effort of evaluating that previous knowledge as rigorously as possible. Besides, we have to be explicit about why we have considered some previous studies and not others. It is reasonable that even after this effort differences of opinion regarding the previous knowledge might remain between different researchers, but at least the reasons for these disagreements would be public and explicit.

Besides, even if the prior probabilities are not agreed upon we can reach an agreement after taking into account the new evidences. With time, as new evidences are accumulated and agreed upon the different prior points of view will converge. To have an efficient research system we have to make an effort of being aware of our prior knowledge and biases and we have to evaluate the new evidences independently of our interests and prior ideas. If we ignore the evidence that contradict our hypotheses and use only the ones that favor us we won’t reach knowledge but opinion.

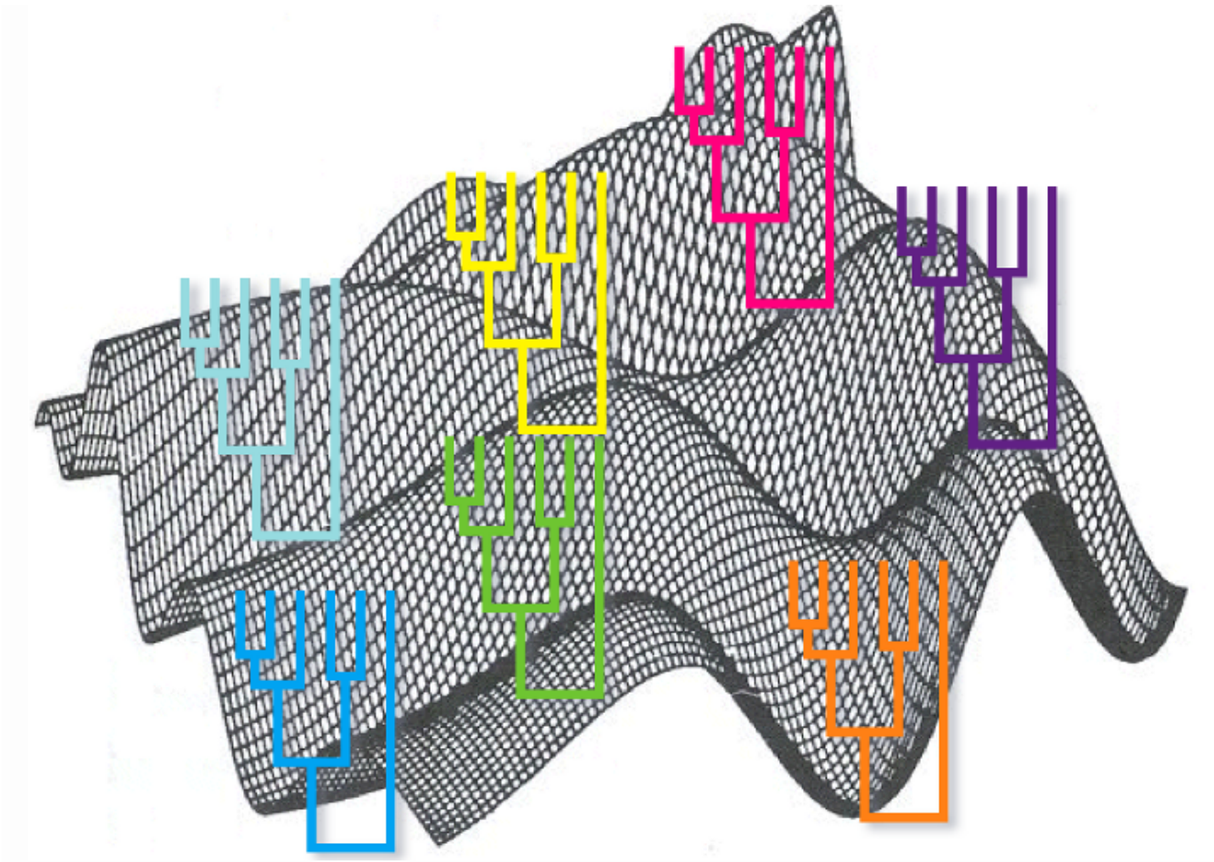
Bayesian methods in phylogeny

In the case of bayesian phylogenetic inferece given the data that we have observed, usually a set of sequences, and some a priori probabilities, we calculate the posterior probabilities for all the possible trees and for all the parameters of the mutations models. The mutation model should be chosen before doing the analyses.

In this method we should evaluate the posterior probabilities for every parameter of every tree. This is computationally impossible for almost any phylogenetic problem. The alternative used is to use Markov chains Monte Carlo (MCMC) methods to sample the parameters and trees. The problem with the MCMC methods is that this chains tend to get stuck in local minima. To solve it the MCMC Metropolis-coupled algorithm is used (MCMCMC).

The most common software in bayesian phylogenetics is MrBayes.

This is the phylogenetic method that is more expensive computationally, but it is regarded as the one that best extracts the phylogenetic information located in a set of observations (usually sequences).



Phylogenetic tree statistical validation

A tree is of not much use if we do not evaluate its statistical significance. A phylogenetic algorithm will always create a phylogenetic tree regardless of the data that we feed it, but that do not imply that the tree is meaningful. We have to evaluate what nodes of a tree can be believe, according to the evidence that we have, and which were generated at random.

In the bayesian methods every node of every tree has a posterior probability associated that we can use to evaluate their confidence, but in the other methods we do not have any direct indication of the reliability of the nodes.

An ideal way to evaluate the reliability of the tree would be to create different trees using independent evidences, for instance sequences from different genes. After building one tree for every gene we could compare which clades are shared by every tree and which are not. The ones shared would be more reliable. The problem with this method is that required different sets of data (although this problem has been alleviated in the genomic era).

A way of generating different alternative trees from one dataset is to do boostrapping. We can do boostraping using any phylogenetic reconstruction method: distance, maximum parsimony and maximum likelihood. The method consists of generating different multiple sequences alignments by replacing columns in the original alignment. For each replica some columns are chosen at random to be replaced and they are replaced by copying other columns. It is a replacement that keeps the number of columns in the alignment invariant. After creating these new alignments we calculate one tree for each of them using the phylogenetic method that we prefer. Finally, we count the number of times in which every clade appears in the boostraped trees and we use that measure as our reliability measure. The clades with high bootstrap values are to be trusted if the assumptions used to construct the tree are true.

There have been a lot of discussion and no consensus about which would be a good threshold to trust a node. It is clear that a clade with a 95% support it is more reliable than a clade with a 50% support, but the intermediate cases are more difficult to evaluate. It is quite common to use 70% as a threshold.

Phylogenetic software

There are different programs to do phylogenies: MEGA, phyml, MrBayes, RAxML and others.