CPS260/BGT204.1 Algorithms in Computational Biology **7 October 2003**

Lecture 12: Genome Sequencing

Lecturer: Pankaj K. Agarwal *Scribe: Philippe Lüdi*

12.1 Sequencing Methods

The early work on sequencing focused on protein sequencing. The amino acid sequence of a peptide has traditionally been determined by *Edman degradation*. This technique involves labeling the N-terminal amino group of a polypeptide and its amino acid components are next cleaved and identified using high-pressure liquid chromatography. Thus, the peptide chain is shortened by one element, as illustrated in Figure 12.1. This procedure is repeated until all amino acids have been identified.

Edman degradation has been routinely in use until the mid 1980s. Since then, recombinant DNA technology has facilitated the process of detecting mRNA and its sequence. This way, the protein sequence can be deduced a lot faster.

Two DNA sequencing methods were developed independently in the 1970s by Maxam and Gilbert and by Sanger.

The *Maxam-Gilbert Method*, developed in the late 1970's, was the first method to determine the sequence of a DNA molecule of up to 500 bp. It involves chemical cleavage at different specific nucleotides of four samples of an end-labeled DNA restriction fragment. This leads to subfragments, which can be separated by gel electrophoresis. Using autoradiography, the labeled fragments can be detected, and the sequence of the original molecule inferred from parallel electrophoretograms of the four samples, as illustrated in Figure 12.2. This technology has the disadvantage of relying on toxic chemicals.

Sanger's Method, developed a few years after the Maxam-Gilbert method, is also referred to as *dideoxy sequencing*. The key element of Sanger's method are 2',3' dideoxynucleoside triphosphates, which lack the hydroxyl group at the 3' position, as depicted in Figure 12.3. Starting from a synthetic 5'-end-labeled oligodeoxynucleotide as a primer, the single stranded DNA molecule to be sequenced is extended by a series of polymerization reactions. Each polymerization reaction involves the normal deoxynucleoside triphosphates (dNTPs) at a high frequency p, and one of the four ddNTPs at a low frequency $1 - p$. So, if one were to sequence the string `TACGTTGCT` using the primer `TAC` and using ddATP dideoxynucleoside triphosphates, the "substrings" produced by the polymerization step would be as follows, since the ddNTPs are incorporated at random and prohibit the molecule from bein further extended:

Substring	Frequency
A	(1-p)
ATGCA	p*p*(1-p)
ATGCAA	p*p*(1-p)

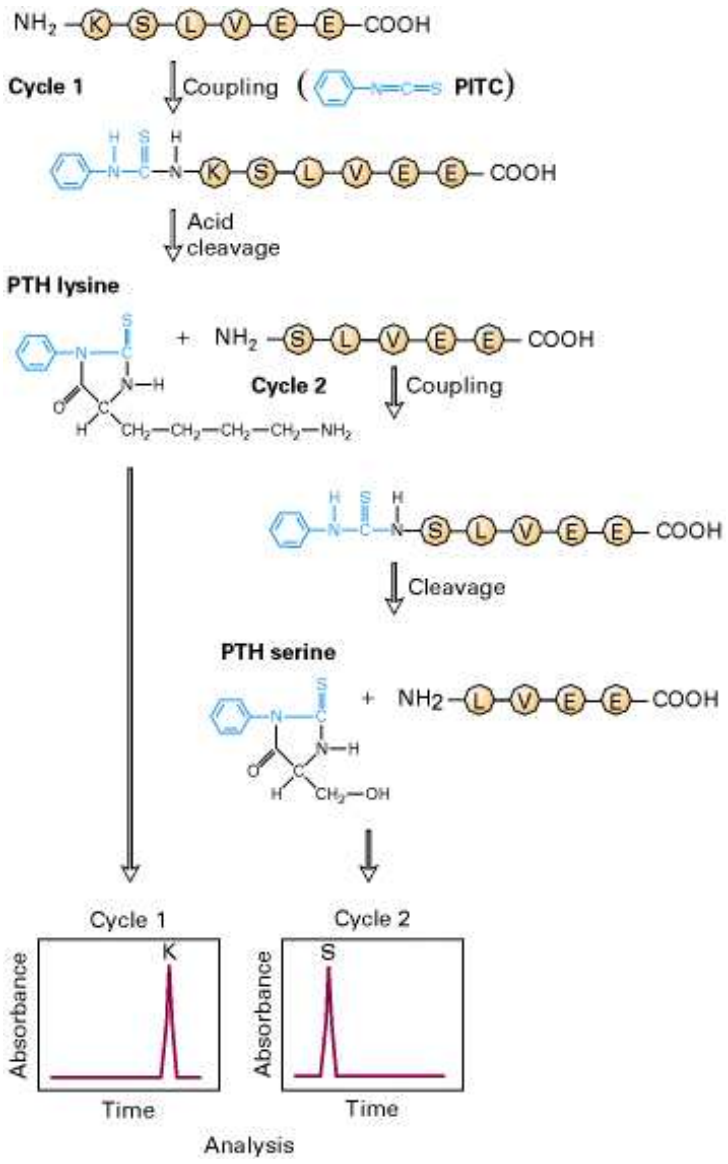

Figure 12.1: Chemical determination of the sequence of a protein by Edman degradation, which involves a repetitive three-step procedure. In the first step, the polypeptide N-terminus is reacted with phenylisothiocyanate (PITC). In the second step, the N-terminal amino acid is cleaved from the polypeptide by acid hydrolysis, yielding the cyclic phenylthiohydantoin (PTH) derivative and a polypeptide that is shorter at its N-terminus by one residue. These two steps are then repeated with the shortened polypeptide. The PTH derivative formed in each cycle is identified by liquid chromatography. (Figure and text taken from Lodish et al. [2000].)

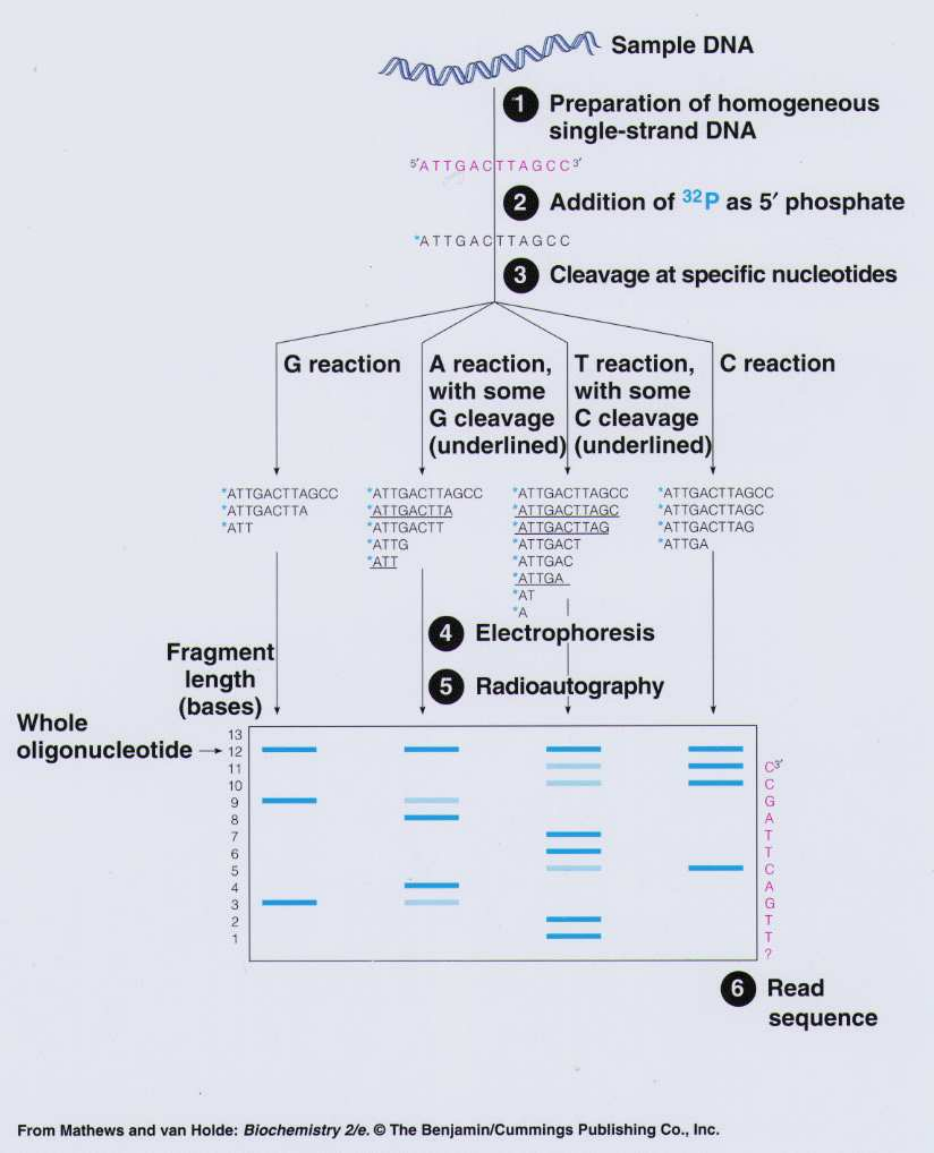

Figure 12.2: The double-stranded fragment to be sequenced is labeled at the 5 ends with P^{32}. The label is removed from one end, and the fragment then is denatured. Four identical samples of the prepared fragment are subjected to four different sets of chemical reactions that selectively cut the DNA backbone at G, G + A, C + T, or C residues. The reactions are controlled so that each labeled chain is likely to be broken only once. The labeled subfragments created by all four reactions have the label at one end and the chemical cleavage point at the other. Gel electrophoresis and autoradiography of each separate mixture yield one radioactive band for each nucleotide in the original fragment, each separated according to their length. Bands appearing in the G and C lanes can be read directly. Bands in the A + G lane that are not duplicated in the G lane are read as A. Bands in the T + C lane that are not duplicated in the C lane are read as T. The sequence is read from the bottom of the gel up. (Text taken from Lodish et al. [2000].)

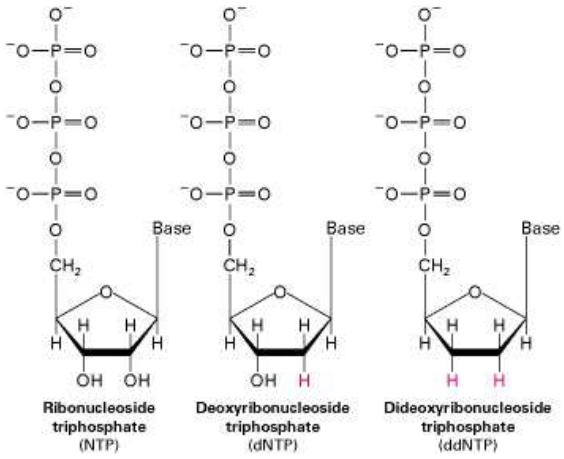

Figure 12.3: Structures of ribonucleoside triphosphate (NTP), deoxyribonucleoside triphosphate (dNTP), and dideoxyribonucleoside triphosphate (ddNTP). (Figure and text taken from Lodish et al. [2000].)

The exact procedure is outlined in Figure 12.4.

A version of Sanger's method is now the most commonly used technique. It involves fluorescently labeled termination products which allow for the reaction mixtures to be pooled and separated by capillary gel electrophoresis. The signal from the four different dyes can then be read at the same time. This process can be very efficiently automated.

Three strategies have been suggested for sequencing the human genome:

- Clone-By-Clone Sequencing
- Sequence Tagged Connector Sequencing
- Whole-Genome Shotgun Sequencing

Clone-by-clone sequencing. In this approach the genome is divided into sections of 150 kb. Each section is fragmented, sequenced, and assembled. Then the sequenced sections are appended to reconstruct the entire genomic sequence. This technique has been chosen by the public Human Genome sequencing effort. A posteriori, it may look unreasonable not to have used whole-genome sequencing. But back when the project was initially started, it was unknown whether this technique would work at all. Furthermore, had the public consortium decided to do shotgun sequencing alone and then discovered that this would not yield the entire sequence, funding to start again would have been impossible to obtain. It is therefore understandable, that the costlier but proven method was chosen. This method is further illustrated in Figure 12.5.

Sequence tagged connector sequencing. This approach is similar to the clone-by-clone approach but involves sequencing the ends of more than half a million BACs (bacterial artificial chromosome). Since the BAC endpoints are cleaved by restriction enzymes, they are not distributed identically over the genome, and

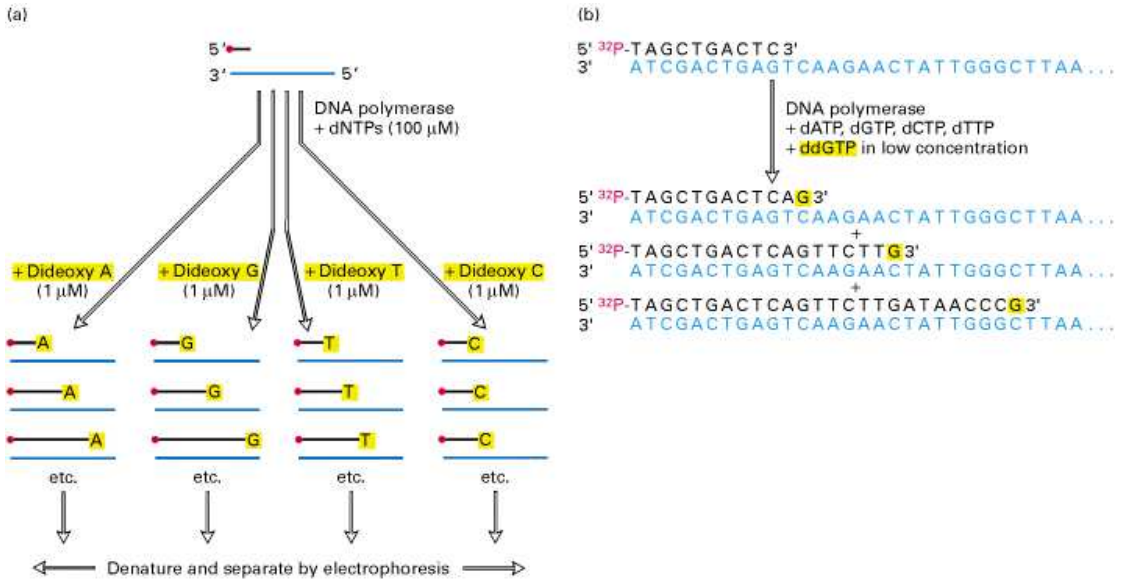

Figure 12.4: Sanger (dideoxy) method for sequencing DNA fragments. (a) A single strand of the DNA to be sequenced (blue line) is hybridized to a 5'-end-labeled synthetic deoxyribonucleotide primer. The primer is elongated in four separate reaction mixtures containing the four normal deoxyribonucleoside triphosphates (dNTPs) plus one of the four dideoxyribonucleoside triphosphates (ddNTPs) in a ratio of 100 to 1. A ddNTP molecule can add at the position of the corresponding normal dNTP, but when this occurs, chain elongation stops because the ddNTP lacks a 3' hydroxyl. In time, each reaction mixture will contain a mixture of prematurely terminated chains ending at every occurrence of the ddNTP (yellow). (Figure and text taken from Lodish et al. [2000].)

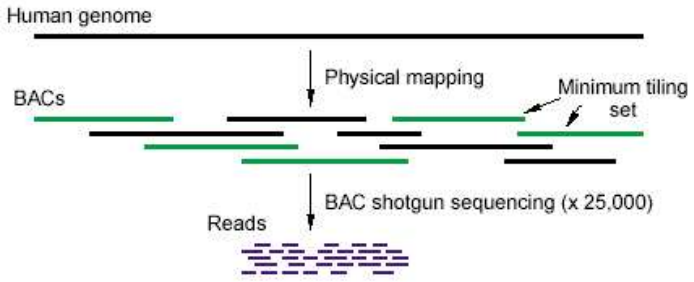

Figure 12.5: The Human Genome Projects two tiered approach. After first fragmenting the genome into large bacterial-artificial-chromosome sized segments, the investigators build a physical map of them. They then select a minimum tiling set of the BACs in the map (shown in green) and shotgun sequence each of these. (Figure and text taken from Myers [1999].)

the finished sequence will thus contain gaps. Additionally, at least $25,00$ BAC libraries are required for this approach, whose construction is involved and expensive. However, no physical map would be required for this approach. Figure 12.6 gives an illustration of this process.

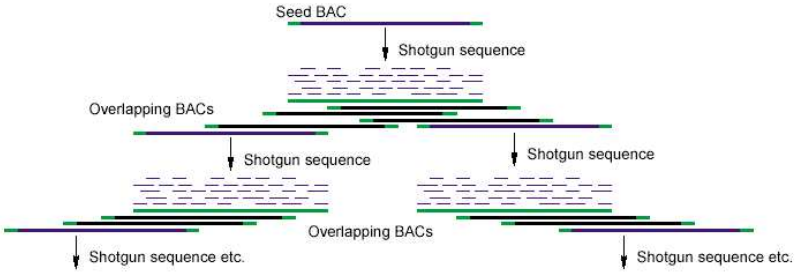

Figure 12.6: Ordered shotgun sequencing. Starting at the top, we shotgun-sequence a selected seed BAC whose sequenced ends are shown in green. Once the entire sequence of this BAC (shown as a solid green line) is revealed, we observe overlaps with a number of end sequences of other BACs in the library. We then shotgun-sequence the left- and rightmost of these (shown with a purple interior). The process continues iteratively, giving a BAC-by-BAC walk across the genome. (Figure and text taken from Myers [1999].)

Whole-genome shotgun sequencing. As presented in Figure 12.7, the entire genomic sequence is fragmented into pieces of clonable length. After amplification in the bacterium *E. coli*, the fragments are sequenced, and assembled. This assembly step is key to the whole-genome shotgun strategy and its critics considered this problem not to be reliably solvable for the Human Genome.

Myers et al. Myers [1999] present the following formalism for whole-genome shotgun sequencing, based on definitions given in Table 12.1. These reflections are motivated by the fact that there are *incomplete sequence coverage*, *sequencing errors* and *unknown orientation* of the DNA molecule sequenced.

Under the (incorrect but useful) assumption of a target sequence with uniform sequence coverage, the fol-

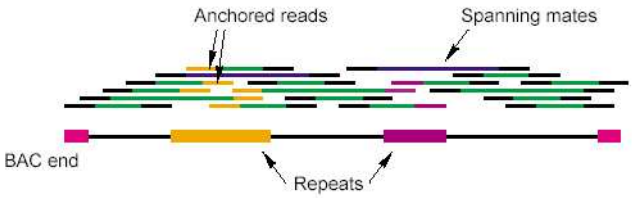

Figure 12.7: Whole-genome shotgun assembly. Mated pairs of fragments are black segments with an intervening green segment connecting them. Given two BAC end sequences shown in red, where for the purposes of illustration we assume there is a gold and purple repeat in the BAC, the problem is to determine the set of mated reads that cover the BAC. Mate pairs that span repeats have their connecting line colored blue, and the reads completely interior to a repeat are given the repeatss color. Such reads are often anchored in the sense that their mate is not in a repeat. (Figure and text taken from Myers [1999].)

lowing probability holds for a DNA fragment:

$$\Pr[\text{left endpoint of a fragment lies in } [x, x+\epsilon]] = \frac{\epsilon}{G},$$

where x and $x+\epsilon, \epsilon > 0$ denote coordinates on a linearly arranged target sequence. Thus, for a set of k fragments, we have a binomial distribution:

$$\Pr[k \text{ fragments have their left endpoints in } [x, x+\epsilon]] = \left(\frac{\epsilon}{G}\right)^k \left(1 - \frac{\epsilon}{G}\right)^{R-k}.$$

If ϵ/G is small, a Poisson approximation can be employed, leading to

$$\Pr[k \text{ left ends lie in the interval } [x - \bar{L}, x]] = \frac{e^{-\bar{c}}\bar{c}^k}{k!}.$$

Thus,

$$\Pr[\text{At least 1 left endpoint lies in } [x - \bar{L}, x]] = 1 - \Pr[\text{No left endpoint lies in } [x - \bar{L}, x]] = 1 - e^{-\bar{c}}.$$

This is the probability of the source strand being covered by at least one read. This requirement in terms of precision allows for the equation to be solved for $\bar{c}$, which in turn leads to the necessary number of sequence reads required. Further, it can be seen that there will be gap-free contigs (DNA segments) of expected length $(\bar{L}/\bar{c})e^{\bar{c}}$, and gaps of expected length $\bar{L}/\bar{c}$.

12.2 Genome Assembly Algorithms

The human genome was sequenced and assembled by two entities, the Public Human Genome Consortium and Celera, with the former not having access to the latter's data, while sequence data from the public effort was available to Celera. Among other differences, it was the challenge posed by sequence repeats that set the two projects apart.

Public Human Genome Consortium (PHGC): only concerned with 150 kbp length sections at a time, so any repeats outside of this section did not present a problem.

G	Length of target sequence
$\bar{L}$	Average length of sequence read
R	Number of sequencing reads in shotgun data set
N	$R\bar{L}$, total number of base pairs sequenced
$\bar{I}$	Average length of a clone inset
$\bar{c}$	N/G, average sequence coverage
$\bar{m}$	$R\bar{L}/2G$, average clone or map coverage

Table 12.1: Definitions.

Celera: repeats had to be correctly assembled, that could span any length of the genome.

Figure 12.8 illustrates the dilemma posed by repeated elements. An overview of the algorithms employed is given below.

Similarities between the two approaches

- Both parties employed a similar strategy for handling uncertain data. First, data was used which the authors were most confident in to build a framework to, later on, place the more unreliable data in. This is the basis of the *Greedy Algorithm*.
- Both use external data such as mRNA, ESTs, YAC-STSs, paired reads and BAC-ends in their assemblies.
- Sequences of contaminants and of vectors used, as well as of repeated elements are masked in the beginning of each algorithm.
- Both *softscreen*, that is, flag repeated elements, as opposed to leaving them out all together. That way certain overlaps can be rejected if the contain different repeats. By *hardscreening* repeats, this would no longer be possible.

Differences between the two approaches.

- The extent to which outside data is used for validation is not the same.
- The metaphors used in Myers' algorithm expresses their confidence in the sequence, unlike Kent et al. whose metaphors refer more to the size of the sequence contigs.
- The two algorithms also differ in terms their input. Myers' algorithm takes a *big bag* of sequence pieces (each about 500bp in length, many of which are mated pairs). Gigassembler, on the other hand, uses a collection of *small bags*, containing output from *predphrap* (Ewing et al. [1998], Ewing and Green [1998]) [1].

[1] see paragraph 12.3

- A key element of Myers' algorithm is his *A-statistic*. This statistic is used to discern the null hypothesis H_0, sequence fragments are present at a certain rate (equivalent to the coverage), versus the alternative hypothesis H_1, that certain sequence fragments are present at higher a rate. To be conservative, a factor of 2 will be employed for H_1.
  It can be found that the likelihood ratio statistic has the following distribution:

$$-2\log(\frac{P(Data \mid H_0)}{P(Data \mid H_1)}) \sim \chi^2.$$

  Myers et al. solely work with

$$\frac{P(Data \mid H_0)}{P(Data \mid H_1)} = Q$$

  and reject H_0 for $Q \geq 10$. For a coverage twice as large, standard results from a Poisson distribution lead to:

$$Q = \frac{\bar{c}^k}{k!}e^{-\bar{c}} \Big/ \frac{(2\bar{c})^k}{k!e^{-2\bar{c}}} = \frac{e^{\bar{c}}}{2^k} = \bar{c}\log_{10}(e) - k\log_{10}(2),$$

  with Poisson arrival rate $\bar{c} = \frac{\rho R}{G}$, where k denotes the number of fragments in a unitig, ρ is the distance between the start of its first fragment and the start of its last fragment, R the number of fragments in a database, and G the estimated genome size.

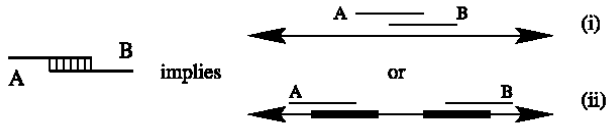

Figure 12.8: True and repeat overlaps. Consider two fragments A and B that overlap as shown at left. There are two possible conclusions depicted at right: (i) the fragments were sampled from overlapping segments of the genome and so belong together in an assembly, a true overlap, or (ii) the overlapping portion is part of a repeated sequence that occurs multiple times in the genome, and the two reads do not belong together, a repeat overlap. Assembly would be a trivial matter if we could divine all the true overlaps; the key objective is to conservatively find true overlaps and to avoid the repetitive ones, especially early in the assembly process. (Figure and text taken from Myers et al. [2000])

The assembly process of any (eukaryotic) genome can roughly be represented as follows:

	Celera	Human Genome Project (Gigassembler)
1. Overlap	- First screen	- First screen
	- Overlapper	- Indexing
2. Layout	- Unitiger	- Layout
3. Consensus	- Establish consensus	- Output from phrap
4. Evidence&Finishing	- PCR	- PCR

Table 12.2: Phases of genome assembly.

Overlapper At the heart of this process lies a big matrix of dimension R, where R denotes the number of sequence reads. Since the matrix is symmetric, there are $\frac{R^2}{2}$ fields to be filled. During this step Celera used a strategy of identifying *high stringency matches* with about 400 bp overlap and allowing for 6% mismatches. Based on this non-overlapping segments can be weeded out and possible final overlaps can be suggested. Myers et al. [2000] were able to compare 32 million pairs of reads per second. Assuming a total of about 3 million sequence reads for *Drosophila melanogaster*, the computations to fill the matrix have probably taken about

$$\frac{4.5 \times 10^1 2}{3 \times 10^7} \cong 2 \text{ days.}$$

An overview of the Celera assembly workflow is presented in Figure 12.9.

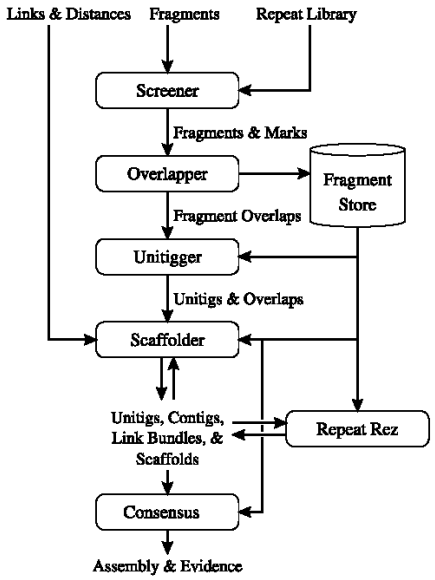

Figure 12.9: Assembly pipeline. From an engineering perspective, sequences of messages flow from one stage to the next. Each stage performs work on its input stream, producing a stream of output messages reflecting its transformational function. The text gives the function of each stage. (Figure and text taken from Myers et al. [2000])

Phrap output Since the output provided by *phrap* represents the results from yet an earlier assembly algorithm, the coverage is, supposedly, less. The quality of the individual bases, however, is better. A key part of the assembly process is *bridge walking*, that is, exploitation of overlaps between BAC's. *Golden Path*, a key word often mentioned somewhat ambiguously in the context of the HGP. The following definition is taken from the Ensembl webpage (*http://www.ensembl.org/Docs/wiki/html/EnsemblDocs/VirtualContigReengineering.html*):

> A golden path is a non redundant set of regions of DNA sequence which make up a chromosome.
> A golden path is a series of start/end points from particular sequences, potentially separated by gaps.

From the UCSC Genome Bioinformatics Site (*http://genome.ucsc.edu/goldenPath/term.html*) is the following quote:

> The sequence produced by combining the information from individual sequenced clones (by creating merged sequenced contigs and them employing linking information to create scaffolds) and positioning the sequence along the physical map of the chromosomes. (Nickname "golden path.")

Golden Path also seems to be the term used to refer to the UCSC Human Genome assembly, as opposed to the NCBI assembly. The following quote is from a University of Washington website (*http://pga.mbt.washington.edu/workshop/refseq_small/tsld003.htm*):

> The "Golden Path" assembly was produced by investigators working in David Haussler's laboratory, located at the University of California, Santa Cruz. This assembly was produced using public data only, and with extensive collaboration from both the National Center for Biotechnology Information (NCBI) and the public genome sequencing centers. Investigators at the UC Santa Cruz used fingerprint-based maps to construct their assembly. Most of the articles published in the journal Nature represent analyses of the Golden Path assembly, including articles in which the names of NCBI investigators appeared. NCBI investigators generated the "NCBI" assembly using only publicly available data, including data deposited in GenBank, the NIH sequence database, by the public genome sequencing centers as well as data submitted to GenBank by independent investigators from around the world. The NCBI assembly is based primarily on sequence overlaps, whereas the UC Santa Cruz group used the fingerprint-based approach to construct their assembly.

12.3 Hands-On Genome Assembly

In addition to the algorithms of Myers et al. [2000] and Kent and Haussler [2001], which are mainly of theoretical interest, the following pieces of software are worth mentioning, as they represent useful tools for the assembly and annotation of smaller eukaryotic genomes or smaller stretches of sequence.

phred, written by Phil Green's group at the University of Washington, is a base calling software, which appears to have a lower error rate than the base calling supplied with sequencing machines, such as the ones manufactured by ABI (Ewing et al. [1998]).

phrap: *phragment assembly program*, or *phil's revised assembly program*; a homonym of *frappe* = French for *swat* — a program for assembling shotgun DNA sequence data. Key features: allows use of entire read (not just trimmed high quality part); uses a combination of user-supplied and internally computed data quality information to improve accuracy of assembly in the presence of repeats; constructs contig sequence as a mosaic of the highest quality parts of reads (rather than a consensus); provides extensive information about assembly (including quality values for contig sequence) to assist trouble-shooting; able to handle very large datasets.

phredphrap: generally used as joint script, which reads in chromatogram files from the DNA sequencer, produces output, mostly in phd format, and continues by assembling the phd files into contigs of ungapped DNA sequences. *consed* (Gordon et al. [1998, 2001], a graphical tool for editing phrap assemblies, is part of

the *phredphrap* package, and is typically used for editing the sequence, finishing the assembly process, and assisting in the annotation procedure. A screenshot of the main consed window is presented in Figure 12.10.

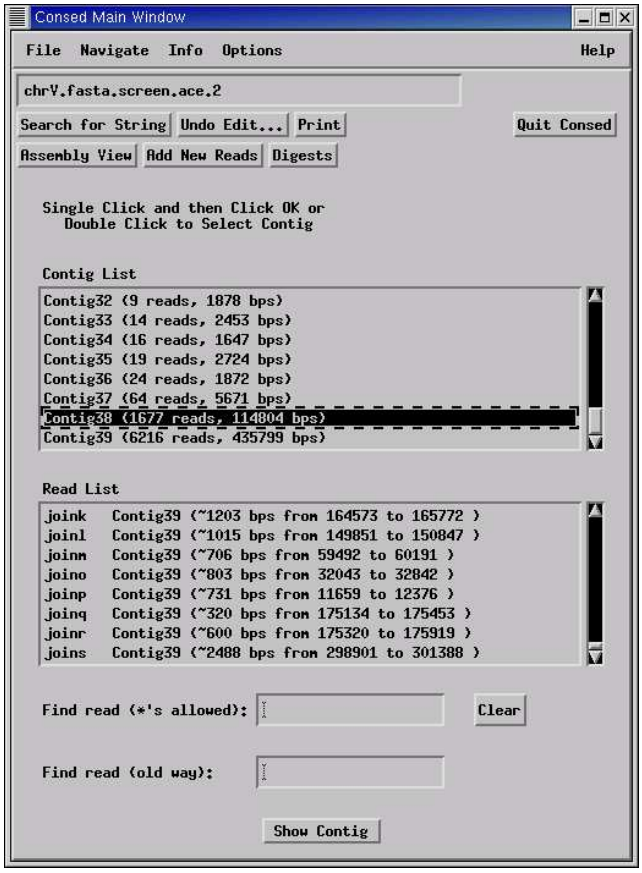

Figure 12.10: Main consed window.

A screenshot of the *Aligned Reads* consed window is depicted in Figure 12.11. In the upper part, the amino acid translations in three frames are presented, followed by the consensus sequence and the sequences of several reads. Note that it is possible to add comments to individual bases, or edit the sequence in other ways, which is then represented by different colours.

The shading of individual bases in *Aligned Reads* consed window, as well as the quality score displayed in the lower part of the window aid in assessing the validity of parts of the sequence. Additionally, by middle-clicking on bases, the respective sequence chromatogram can be brought up, as depicted in Figure 12.12.

A fourth element of the *phredphrap* package is *Autofinish* [Gordon et al., 2001]. It offers the following functionality (quoted from *http://www.phrap.org/consed/consed.html*):

- Figure out how contigs are ordered and oriented
- Close gaps

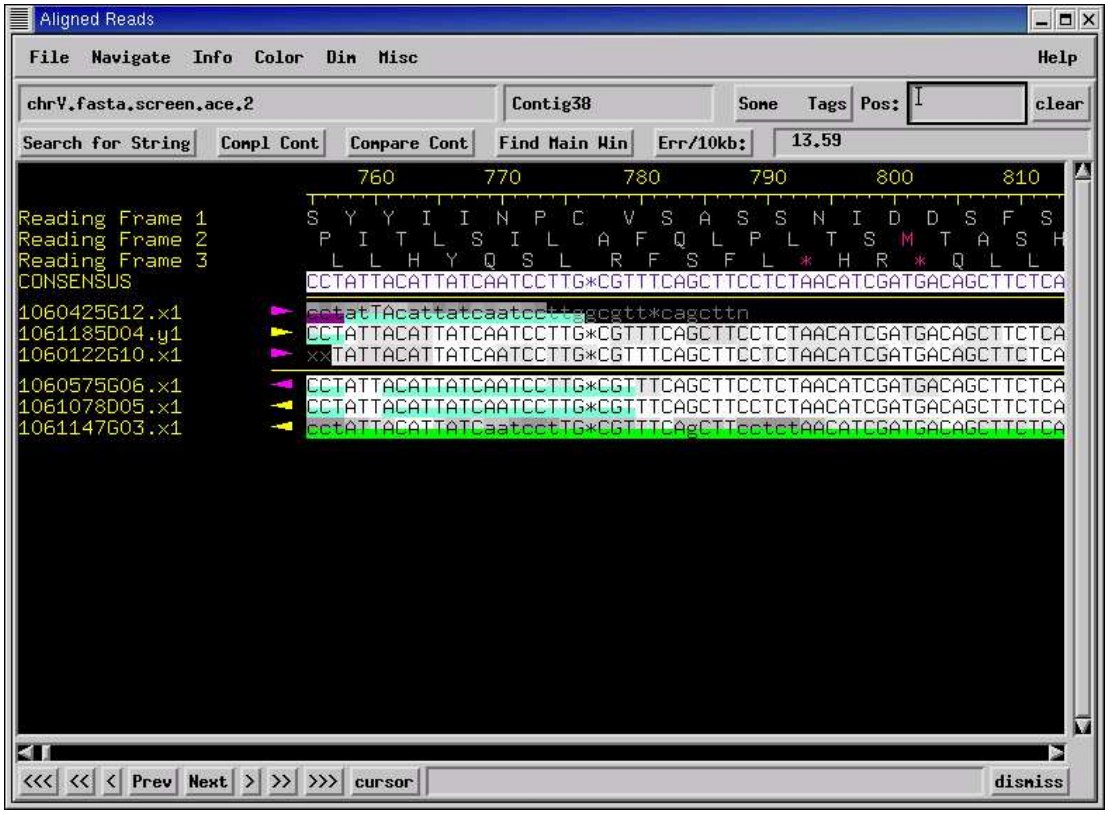

Figure 12.11: Aligned reads window.

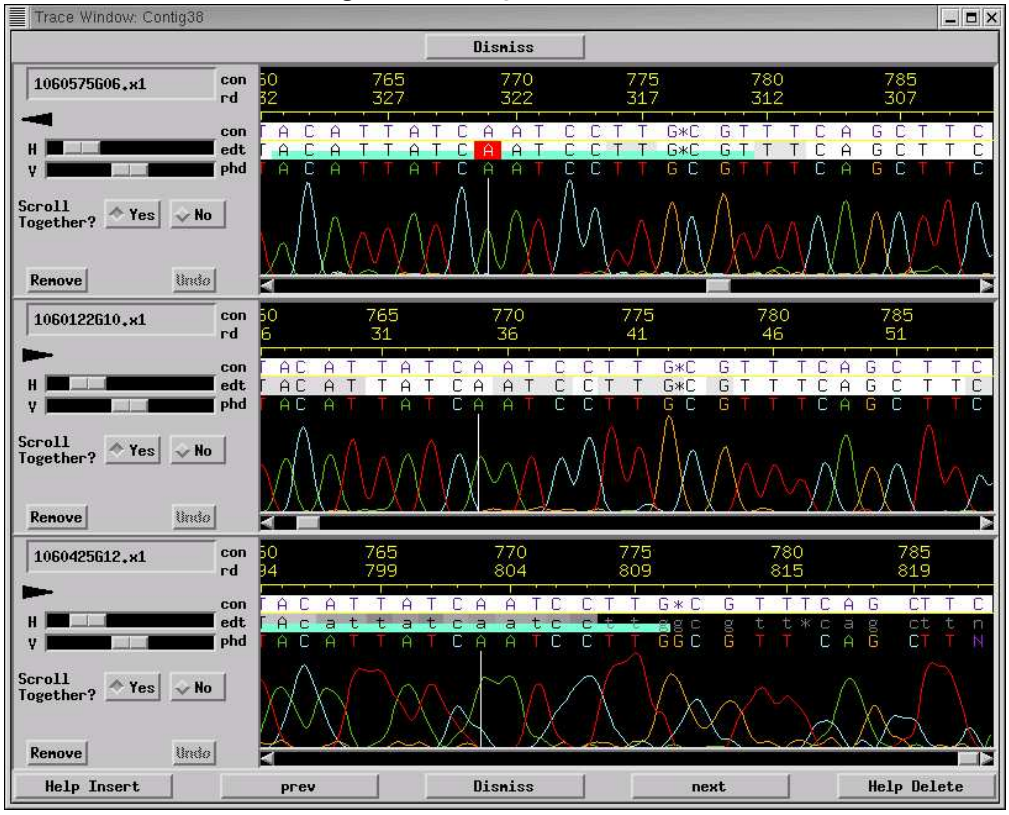

Figure 12.12: Trace window.

- Improve the error rate
- Cover every base by reads from at least 2 different subclones

Autofinish will suggest any or all of the following types of reads:

- Forward universal primer terminator reads
- Reverse universal primer terminator reads
- Custom primer reads with subclone template
- Custom primer reads with whole clone template
- Minilibraries
- PCR

Recent alternative assembly algorithm. Even though parts of the Bioinformatics community consider the whole-genome shotgun assembly problem to be solved, research in this direction continues. *Arachne* [Batzoglou et al., 2002], is a recently developed assembly program. The authors describe their algorithm as follows:
"Myers and colleagues produced a sophisticated assembly program, called the Celera assembler (Myers et al. 2000), based on various layout algorithms previously developed by Myers (1995). ARACHNE shares some significant similarities with the Celera assembler, most notably in the algorithms for merging reads into contigs up to the boundaries of repeats (Myers 1995). However, the two assemblers also have many significant differences. The Celera assembler screens for predefined repeats; ARACHNE does not and instead uses k-mer frequencies to identify repeats. ARACHNE uses sorting of k-mers to detect overlaps; the initial version of the Celera assembler employed a different approach (although the program has been subsequently revised (G. Myers, pers. comm.). Like PHRAP (Green 1994) and CAP3 (Huang and Madan 1999), ARACHNE creates, refines, and evaluates read alignments using quality scores; the Celera assembler does not use quality scores subsequent to trimming and prior to consensus. ARACHNE uses error correction to more accurately evaluate read overlaps. Both assemblers detect potential repeat boundaries, merge reads up to these boundaries (as in Myers 1995), and use read density to detect repeat contigs. ARACHNE also detects repeat contigs via conflicting links. Moreover, it uses paired pairs and other techniques to produce longer contigs during layout. Both ARACHNE and the Celera assembler require at least two links to order a pair of contigs in a supercontig, but apart from that they use different algorithms to build supercontigs and fill in gaps within supercontigs."

References

S. Batzoglou, D. B. Jaffe, K. Stanley, J. Butler, S. Gnerre, E. Mauceli, B. Berger, J. P. Mesirov, and E. S. Lander. Arachne: a whole-genome shotgun assembler. *Genome Research*, 12(1):177–189, 2002.

B. Ewing and P. Green. Base-calling of automated sequencer traces using phred. ii. error probabilities. *Genome Research*, 8(3):186–194, 1998.

B. Ewing, L. Hillier, M. C. Wendl, and P. Green. Base-calling of automated sequencer traces using phred. i. accuracy assessment. *Genome Research*, 8(3):175–185, 1998.

D. Gordon, C. Abajian, and P. Green. Consed: A graphical tool for sequence finishing. *Genome Research*, 8 (8):195–202, 1998.

D. Gordon, C. Desmarais, and P. Green. Automated finishing with autofinish. *Genome Research*, 11(4): 614–625, 2001.

W. J. Kent and D. Haussler. Assembly of the working draft of the human genome with gigassembler. *Genome Research*, 11(9):1541–1548, 2001.

H. Lodish, A. Berk, S. L. Zipursky, P. Matsudaira, D. Baltimore, and J. Darnell. *Molecular Cell Biology*. Scientific American Books, Inc., 4 edition, 2000.

E. W. Myers. Whole-genome dna sequencing. *IEEE Computational Engineering and Science*, 3(1):33–43, 1999.

E. W. Myers, G. G. Sutton, A. L. Delcher, I. M. Dew, D. P. Fasulo, M. J. Flanigan, S. A. Kravitz, C. M. Mobarry, K. H. Reinert, K. A. Remington, E. L. Anson, R. A. Bolanos, H. H. Chou, C. M. Jordan, A. L. Halpern, S. Lonardi, E. M. Beasley, R. C. Brandon, L. Chen, P. J. Dunn, Z. Lai, Y. Liang, D. R. Nusskern, M. Zhan, Q. Zhang, X. Zheng, G. M. Rubin, M. D. Adams, and J. C. Venter. A whole-genome assembly of drosophila. *Science*, 287(5461):2196–2204, 2000.