CHAPTER 7
Expressed sequence tags

Arthur Gruber

1. INTRODUCTION

Expressed sequence tags (ESTs) are short sequence reads, typically within the range of 100–700 bp (see *Fig. 1*), obtained from randomly selected cDNA clones. The concept was first introduced as a cost-effective approach for the rapid discovery and characterization of expressed genes (1). ESTs are often generated by single-pass sequencing of cDNA clones from one or both ends, usually covering only a part of the transcript sequence, and are relatively prone to error. Despite this latter feature, EST sequencing represents a mainstream methodology for gene surveying. Even nowadays, when whole genome sequences are available for many organisms, ESTs still play an important role in gene identification, transcript mapping, and description of the transcriptional activity of a tissue/cell type. Furthermore, ESTs may represent a very important body of evidence for gene prediction, and an abundant resource of molecular markers for physical mapping (2). Another envisaged application of ESTs is the quantification of gene expression, as the abundance of sequence reads representing each transcript may reflect the steady-state levels of these transcripts in a tissue or cell type (3). Finally, ESTs may

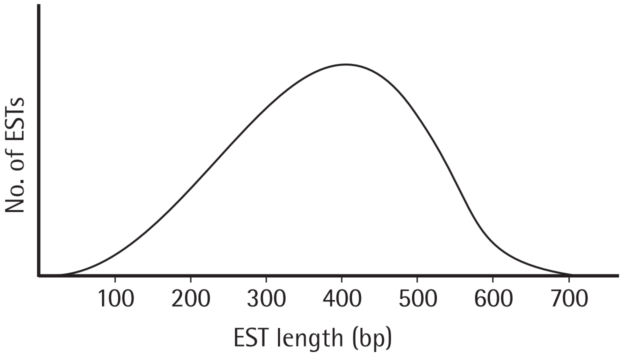


Figure 1. Distribution of EST reads according to the sequence length.
A typical EST sequencing project presents read lengths ranging from 100 to 700 bp, with most reads falling around 400 bp. Nevertheless, values may vary depending on the sequencing equipment, reagents, and protocols utilized.

Bioinformatics: ***Methods Express*** (Paul H. Dear, ed.)

provide reagents for downstream applications such as microarray analysis and immunoscreening of potential protective antigens. This chapter will cover some important aspects of EST analysis, including EST clustering, redundancy estimation, automated processing pipelines, and EST database searching. In addition, some important but often neglected methodological details will be also discussed. For additional literature on EST data production and analysis, the reader is advised to consult some reviews (4-9). The role of ESTs in gene prediction and annotation is touched on in this chapter, but the reader is referred to Chapter 4 for a more extensive discussion of gene prediction.

1.1 EST library construction and sequencing

In order to apply effectively the bioinformatics tools that will be covered in this chapter, the reader needs to understand how EST sequences are generated, as this will determine their strengths and weaknesses. We will, therefore, briefly outline the wet-lab side of EST production.

Different methods are currently available for cDNA library construction and will be discussed here in light of some implications on coverage within transcripts. However, it is beyond the scope of this chapter to present the different methods for EST library construction in depth, and the reader is advised to consult some specific reviews (9, 10). *Fig. 2*(*a*) depicts the most common methods for cDNA synthesis. Purified mRNA is used as a template for reverse transcriptase using either oligo(dT) as a primer for the first-strand synthesis or, alternatively, random hexamer primers (9, 10). After nicking the RNA–DNA hybrid with RNAse H, second-strand synthesis proceeds using the RNA fragments as primers and DNA polymerase I for the extension. Another approach for cDNA library construction is the ORESTES (ORF ESTs) method. The protocol is based on the construction of multiple mini-libraries (11, 12) using low-stringency reverse transcriptase polymerase chain reaction (RT-PCR) amplifications with arbitrary primers. Each mini-library, constructed with a particular primer, results in a heterogeneous population of amplified cDNA products, which correspond to a subset of the expressed gene profile. As each oligonucleotide acts as both forward and reverse primer, the distribution of the amplification products is biased towards the central part of the transcript (11). Whatever the synthesis method chosen, it is good practice to size-fractionate the cDNA fragments to avoid cloning very small inserts.

Conventional libraries can be constructed with specific adapters that permit unidirectional cloning (13). In this case, the researcher can choose which end of the transcript cDNA will be sequenced. On the other hand, because ORESTES libraries employ arbitrary primers, no directionality can be obtained. The synthesis method and the choice of the fragment end to be sequenced have a direct implication on the sequence coverage within transcripts (see *Fig. 2b*). A 5′-end sequencing results in a higher proportion of clones covering the coding regions of the transcripts, which is quite convenient for gene discovery projects (13). Conversely, 3′-end sequencing results in an extensive coverage of the 3′ end of the transcripts. As

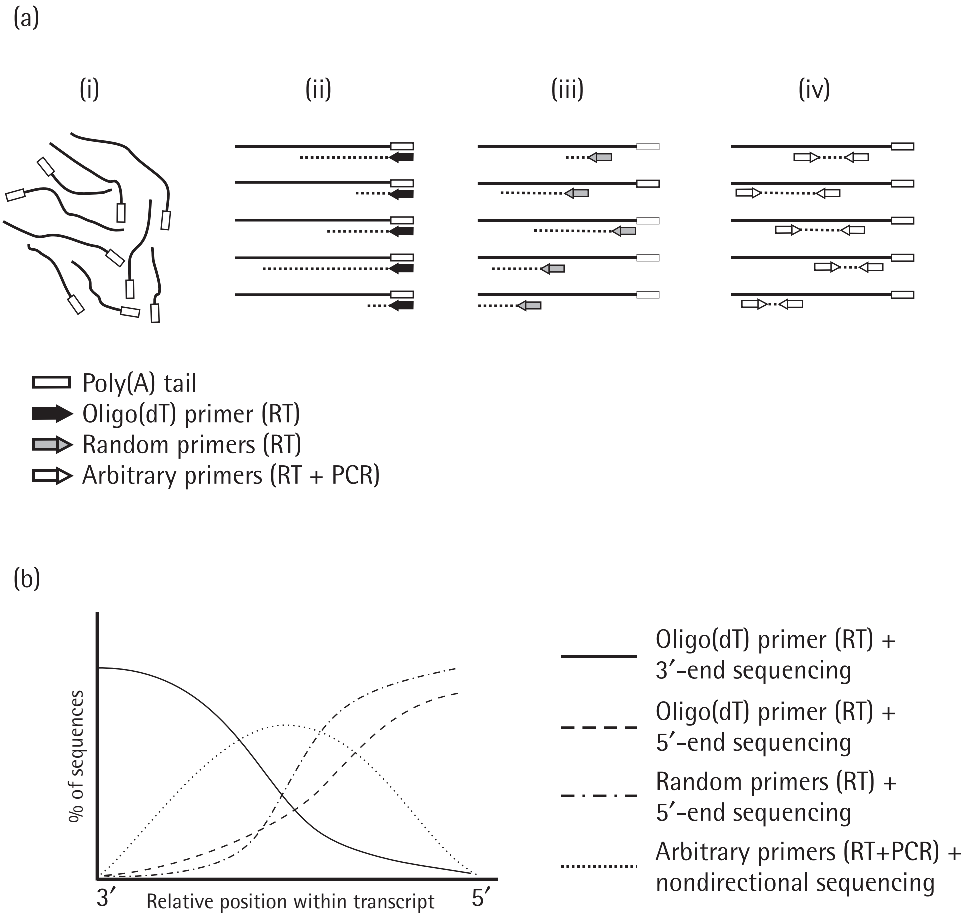

(a)
(i)
(ii)
(iii)
(iv)
Poly(A) tail
Oligo(dT) primer (RT)
Random primers (RT)
Arbitrary primers (RT + PCR)
(b)
% of sequences
3′
Relative position within transcript
5′
Oligo(dT) primer (RT) + 3′-end sequencing
Oligo(dT) primer (RT) + 5′-end sequencing
Random primers (RT) + 5′-end sequencing
Arbitrary primers (RT+PCR) + nondirectional sequencing

Figure 2. Different methods for generating cDNA libraries.

The scheme (*a*) shows the first-strand synthesis approaches most commonly used to construct cDNA libraries. The mRNA (*i*) is purified from the total RNA by affinity chromatography using oligo(dT)-linked resins or paramagnetic beads. The most conventional method *(ii)* uses an oligo(dT) primer that anneals to the poly(A) tail and a reverse transcriptase (RT) that catalyzes the cDNA synthesis (9, 10). Daughter strands are of varying lengths and the 5′ ends of the transcripts are usually poorly represented if 3′-end sequencing is carried out (*b*, solid line). Better transcript coverage may be attained by sequencing the 5′ ends (*b*, dashed line). A variation of this method uses random primers instead of the oligo(dT) primer (*iii*). They consist of fully degenerate hexamers (dN$_6$), sometimes linked to an anchor sequence at the 5′ end. This method yields a better coverage of the 5′ end of the transcripts, especially if directional cloning followed by 5′-end sequencing is used (*b*, dash–dot line). In both cases (*a*, *ii* and *iii*), second-strand synthesis is performed by conventional methods, employing RNAse H for nicking the RNA strand of the DNA–RNA hybrid and DNA polymerase I to replace the RNA segments by DNA (not shown here). An alternative approach, termed ORESTES (11, 12), utilizes an arbitrary primer for both RT reaction and subsequent PCR amplification (*d*). As the same oligonucleotide acts as both the forward and reverse primer, distribution of the amplified products is biased to the central part of the transcripts (*b*, dotted line).

these regions are much more variable than the coding regions, they can be used for the unambiguous identification of the transcripts, thus allowing quantitative gene expression profiling (14–16). In fact, this characteristic has been used in the construction of the UniGene database (14), in which the clusters are anchored by unique 3′-untranslated regions (3′UTRs). In addition, because intronic sequences are rare in 3′UTRs, such ESTs can also be used as sequence-tagged sites (STSs) for genome mapping (17). ORESTES reads, on the other hand, are biased towards the central part of the transcripts (11, 12), which prioritizes the protein-coding information. This aspect makes this method a good choice for gene discovery projects and, furthermore, generates complementary data to 5′- and 3′-end sequencing efforts.

1.2 Representation: normalized and subtracted libraries

The *transcriptome* of an organism can be defined as its complete repertoire of transcripts, including splice variants. Gene expression in all organisms varies over time (for example, during development or in response to changing conditions) and, in complex eukaryotes, from one cell type to another. Therefore, the transcriptome of a particular tissue, sampled at a particular time, will only be a subset of the complete transcriptome of the organism. Of course, much of the value of EST analysis lies in comparing the transcriptomes of different cell types, at different developmental stages or under different conditions.

In principle, a nonbiased cDNA library should faithfully represent the transcriptome of the cell or tissue that it was derived from, provided that a sufficient number of reads has been obtained. However, because some mRNAs are highly expressed whilst many others are only found in tiny amounts, this latter class is under-represented in any cDNA library. If the goal of the EST sequencing project is to obtain a comprehensive survey of the transcriptome, then a very large number of reads would have to be sequenced, making this approach too expensive. In order to avoid missing rare transcripts whilst maintaining a cost-effective approach, normalization techniques have been devised to decrease the relative representation of abundant transcripts whilst increasing that of rare transcripts. The commonest normalization method relies upon reassociation kinetics (9, 18–20): the cDNA is denatured and, because of second-order kinetics, abundant cDNAs tend to renature more quickly than scarce ones. Hydroxyapatite chromatography can then be used for selective purification of the remaining single-stranded molecules, which will be relatively enriched for the scarce transcripts. ORESTES also normalizes the cDNA representation (11, 12) by employing a high number of amplification cycles, so that abundant transcripts attain saturation in the early steps of cycling, whereas rare transcripts continue to be amplified during the later cycles.

Another methodology to change the representation of cDNA libraries is subtractive hybridization (21, 22). This method is used to reduce the representation of transcripts already surveyed in previous libraries, as well as to enrich for sequences that are differentially expressed among specific tissues, cell types, etc.

The cDNA library that will be the target of subtraction (the 'tester' population) is denatured and hybridized to another library that is present in excess (the 'driver' population). Fragments that are common to both populations anneal to each other, whereas the tester-specific products will remain single stranded. The purification of these products can be achieved by either hydroxyapatite chromatography or avidin–biotin binding.

2. METHODS AND APPROACHES

2.1 Overview

EST analysis is a complex multiple-step process that makes use of several distinct programs. In this section, we will give a brief introduction to the theory behind each method and present some specific protocols covering worked examples of widely used methods. As no generic step-by-step recipe can fulfill all specific requirements and characteristics of different EST sequencing projects, we have chosen to offer some typical protocols in a tutorial-like presentation where the reader will be able to adapt the techniques to fit his or her needs with only minor changes. We recommend using a UNIX/Linux machine connected to the Internet through a broadband connection. In our hands, a PC-based server running Linux is the most cost-effective platform and enables one to run analyses of even relatively large numbers of ESTs. Apple Mac OS X 10.x is another recommended choice. For very large datasets (hundreds of thousands to millions of ESTs), a more powerful workstation or a cluster of PCs is recommended. We are assuming that a Perl interpreter (http://www.perl.org [7.1]) has previously been installed on your server, as well as Java Platform 2 (http://java.sun.com/j2se/ [7.2]). Other specific programs are listed in each protocol and must be installed on your server before running the tutorials. All software chosen for the protocols is open source or free for nonprofit academic use (other policies may apply for commercial use), and is available on the Internet or on request to the authors. Each protocol lists the required software, corresponding publication (where available), sources for download, and/or author's contact details. Software installation is relatively easy, even for novice users, but may require administrator privileges. Please contact your local server administrator in case you need any help in installing the necessary programs.

Example datasets for this chapter are provided on the book's web site. Because of the large number of subdirectories and files involved, the complete dataset is provided as a single compressed file; *Protocol 1* describes how to install the example datasets.

Protocol 1

Installing the example datasets

1. Download the dataset file protocols.tar.gz [7.3] from the All _Protocols folder for this chapter on the book's web site.
2. Decompress the file with the following UNIX/Linux command: '`tar zxvf protocols.tar.gz`'. If this command does not work (depending on your system's configuration), then try the following command:

   ```
   gzip -dc protocols.tar.gz | tar xvf -
   ```

   Either of these commands will extract all files and subdirectories within a directory named protocols.
3. Move your protocols directory to the selected location on your server disk. You may need to have administrator privileges to do this:

   ```
   mv protocols /selected_directory
   ```

   There are four subdirectories within the protocols root directory: Protocol_2, Protocol_4, Protocol_5, and Protocol_6. The required subdirectories and files will be listed in each of the protocol sections below.

2.2 EST databases

In this section, we will review some of the EST resources available and describe how to retrieve data from two of them – dbEST and UniGene.

2.2.1 dbEST

Genbank's EST database (dbEST) is a publicly available repository useful for gene discovery and comparative gene expression studies (17, 23, 24). It constitutes the EST division of Genbank (25) and corresponds to the largest fraction (48%) of entries (source: NCBI web page http://www.nlm.nih.gov/ [7.4]). As of October 2006, dbEST (release 100606) comprised more than 38 million entries from over 1200 organisms, although about half of the entries come from only eight organisms (see *Table 1*). In *Protocol 2*, we will see how to download an organism-specific set of ESTs.

Table 1. Top 20 organisms represented on dbEST (release 100606)

Organism	Number of ESTs
Homo sapiens (human)	7 893 983
Mus musculus + *domesticus* (mouse)	4 720 064
Oryza sativa (rice)	1 188 565
Zea mays (maize)	1 143 728
Bos taurus (cattle)	1 137 353
Danio rerio (zebrafish)	1 134 553
Xenopus tropicalis	1 044 182
Rattus norvegicus + sp. (rat)	871 144
Triticum aestivum (wheat)	855 066
Ciona intestinalis	686 396
Sus scrofa (pig)	623 929
Arabidopsis thaliana (thale cress)	622 973
Gallus gallus (chicken)	599 141
Xenopus laevis (African clawed frog)	537 424
Drosophila melanogaster (fruit fly)	514 545
Hordeum vulgare + subsp. *vulgare* (barley)	437 321
Canis familiaris (dog)	365 909
Glycine max (soybean)	359 151
Caenorhabditis elegans (nematode)	346 064
Pinus taeda (loblolly pine)	329 469

Source: NCBI (http://www.ncbi.nlm.nih.gov/ 7.5).

Protocol 2

Downloading an organism-specific set of ESTs from dbEST

1. Point your web browser to the NCBI's site at http://www.ncbi.nlm.nih.gov/ 7.5.
2. Select the **Nucleotide** option (default is **All Databases**) on the **Search** selection box and type '`Plasmodium falciparum [organism]`' in the blank form. An organism name followed by the '`[organism]`' tag will limit the sequence retrieval to this particular organism (see Chapter 1). Click on the **Go** button to retrieve the records.
3. The page will now display the total number of records retrieved (see **All:** approximately 43 000 records were retrieved at the time of writing, although this is of course likely to increase over time) and a summary of the first 20 nucleotide records of *P. falciparum.* Click on the **Search** selection box again and select **EST** (a subset of the nucleotide database) from the pull-down menu. Click on the **Go** button to retrieve the records[a]. The total number of entries (about 21 000 at the time of writing) will be displayed on the top of the record list.
4. Next to **Display** towards the top of the screen, change the setting from **Summary** (the default) to **FASTA** format option. Then select **File** from the **Send to** pull-down menu. The program will ask for confirmation and then the download process will start, creating a large file called sequences.fasta.

5. If you want to confirm that the file is complete and contains all records retrieved from the database (see step 3), you can type the following UNIX command:

```
grep ">" sequences.fasta | wc -l
```

This command extracts all lines displaying a '>' character (present in each FASTA header) and counts the total number of lines obtained: this is a simple way of counting how many sequences are present in the file. The number should be equal to the number of entries displayed in step 3.

6. Rename the downloaded file as P_falciparum.fasta. On UNIX systems, this is done using the command:

```
mv sequences.fasta P_falciparum.fasta
```

7. In the Protocol_2 directory, we provide a P_falciparum.fasta[7.6] file that was downloaded using the above method in October 2006 and contains 21 349 sequences in FASTA format. By the time you download yours, the number of sequences will probably be larger, as new entries are continuously being incorporated into dbEST.

Note

[a] If you prefer, you can combine steps 2 and 3 into one: you can specify both the organism and EST subset by typing '`Plasmodium falciparum [organism] AND gbdiv_est [PROP]`' in the blank query window.

2.2.2 TIGR Gene Indices (TGI)

TGI, now hosted at the Dana-Farber Cancer Institute (http://compbio.dfci.harvard.edu/tgi/[7.7]), is a database of clustered ESTs from many eukaryotic organisms (26). GenBank coding sequences from genomic and mRNA sequences are clustered by pairwise alignments using a modified version of MEGABLAST (27) and the clusters are then assembled using CAP3 (28). The final result is a set of tentative consensus sequences representing unique transcripts. The tentative consensus sequences are integrated with annotation data into a relational database and can be queried using text and BLAST searches.

2.2.3 STACKdb

STACKdb (29, 30) (http://www.sanbi.ac.za/Dbases.html[7.8]) is a database of clustered ESTs hosted at the South Africa National Bionformatics Institute (SANBI). Clustering is performed with D2_CLUSTER (31) and STACKPACK (30) software. EST sequences are clustered, assembled, and then submitted to a post-assembly analysis protocol in which clusters spanning different regions of a transcript are identified and grouped. Single-nucleotide polymorphisms (SNPs) and splicing variants are also identified.

2.2.4 UniGene

UniGene (http://www.ncbi.nlm.nih.gov/UniGene[7.9]) is an automatic system for partitioning GenBank sequences into a nonredundant set of gene-oriented clusters (14). Each cluster corresponds to a unique transcript, but, unlike TGI and STACKdb, UniGene clusters are not assembled and, therefore, no consensus sequences are available. UniGene clusters also provide some important cross-information such as the tissue types where gene expression was observed and corresponding map locations. UniGene is integrated (32) with other databases such as the IMAGE Consortium (33), a public domain resource of arrayed cDNA libraries, and sequence, map, and expression data.

Protocol 3 illustrates how to search for UniGene clusters and retrieve information of EST clusters, protein similarity and annotation, gene expression, mapping position, and sequence data. For this purpose we will use, as the query, glucagon (a polypeptide hormone secreted by the alpha cells of the pancreas islets of Langerhans in response to hypoglycemia).

Protocol 3

Using UniGene

1. Point your web browser to the NCBI's UniGene site at http://www.ncbi.nlm.nih.gov/UniGene [7.9].
2. Into the text field, next to **Search Unigene for**, enter '`glucagon[All Fields] AND ("Homo sapiens"[Organism])`', taking care over the brackets. This query will retrieve information on the glucagon gene, restricted to human sequences. (For more information on querying NCBI databases, refer to Chapter 1, or follow the **Query Tips** link from the web page). Click **Go**.
3. Select the glucagon gene by clicking on the **Hs.516494** link.
4. Now a list of information related to glucagon is displayed. We will start by exploring the link for the Swiss-Prot entry. For this option, click on the **sp:P01275** link in the 'SELECTED PROTEIN SIMILARITIES' section.
5. The new page presents a very comprehensive annotation on the protein, including pertinent bibliographic data, a typical Swiss-Prot description of the protein function, site of production, and pharmaceutical information, plus the corresponding amino acid sequence.
6. Go back to the previous page and click on the **Expression profile** link in the 'GENE EXPRESSION' section. The newly opened page displays a table presenting the gene expression deduced from the analysis of EST counts of different tissues. As expected, pancreas is by far the predominant expression site.
7. Return to the former page and, still in the 'GENE EXPRESSION' section, click on the **GEO profiles** link. This will take you to the Gene Expression Omnibus (GEO) database, a repository of gene expression profiles that includes experimental data of microarray, serial analysis of gene expression (SAGE), and mass spectrometry proteomic experiments. A list of GEO records will be presented, with the respective expression profile data graph on the right.
8. Choose the record corresponding to the experiment with 'Normal tissues of diverse types (SHCN)'. When writing this text, the experiment corresponded to the **GDS1086 record**, but it may have changed by the time you access the site. Click on the expression graph on the right. The graph displays the analysis of glucagon expression in dozens of physiologically normal tissues obtained from various sources. As expected, pancreas tissue presents the highest expression of this gene.

9. Go back to the previous page (listing all of the experiments) and click on the **Record** link for this experiment. This will take you to the corresponding microarray data and the clustering analysis.
10. Go back to the previous page and back once more to the glucagon entry page (this is the page you reached in step 3). Look at the information available at the 'MAPPING POSITION' section (scroll down). It reports that the glucagon gene is located at chromosome 2, with the cytogenetic locus 2q36–q37.
11. Below this information (under 'SEQUENCES'), the page displays a list of all mRNA and EST sequences related to this UniGene cluster. Sequences can be retrieved on an individual basis by clicking on the respective links or, alternatively, can easily be downloaded in a batch by clicking on the **Download sequences** button at the bottom of the page.
12. Repeat the whole analysis, now using the human hemoglobin beta-chain gene. Click on the **Expression profile** link. The expression profile page will show that blood cells, bone marrow, and muscle are the preferential expression sites. Compare the results with those obtained for the glucagon gene.

2.3 Automated EST pre-processing pipelines

Proper multi-step pre-processing of EST raw data is necessary before one can proceed with clustering and assembly (6) or with downstream annotation. *Fig. 3* displays a typical EST processing pipeline scheme. First (*Fig. 3a*, *b*), the trace files, if available, are submitted to a base caller and quality evaluation program such as PHRED (34, 35). In this step, the nucleotide sequence is extracted and confidence values (also known as 'PHRED values') are then ascribed to all bases. Low-quality sequence can then be filtered out. If only sequence data is available on a public database, then no confidence values can be ascribed and quality filtering cannot be done. Secondly (*Fig. 3c*, *d*), undesirable parts of the sequence are 'masked' by replacing them with N or X characters, which are disregarded by most clustering and assembly programs. For example, simple sequence repeats (SSRs) can be identified and masked by programs such as DUST (36) and TANDEM REPEATS FINDER (37), and the sequences of vectors and/or primers used in making the EST library can be identified by comparison with specific databases. Poly(A) tracts should also be identified and masked, especially when dealing with 3′-end directional libraries. All of these masking steps are essential to avoid unrelated ESTs being grouped into the same cluster/contig. Thirdly (*Fig. 3e–h*), contaminant sequences must be identified and filtered out. Contaminants may comprise endogenous sources such as ribosomal sequences and transcripts derived from organelles such as mitochondria or plastids, and heterologous sources such as bacteria. Contaminant filtering is done by pairwise alignment of the ESTs against databases containing likely contaminant sequences; a rule of thumb is to perform the filtering steps using the smaller databases at the earlier steps of the pipeline and the larger databases at the later steps. As positive matches are identified and discarded, the overall number of reads that have to be processed by the later slower steps is reduced, thus decreasing the overall pipeline processing time. Finally, the resulting 'clean' EST reads can be assembled into clusters believed to represent distinct transcripts.

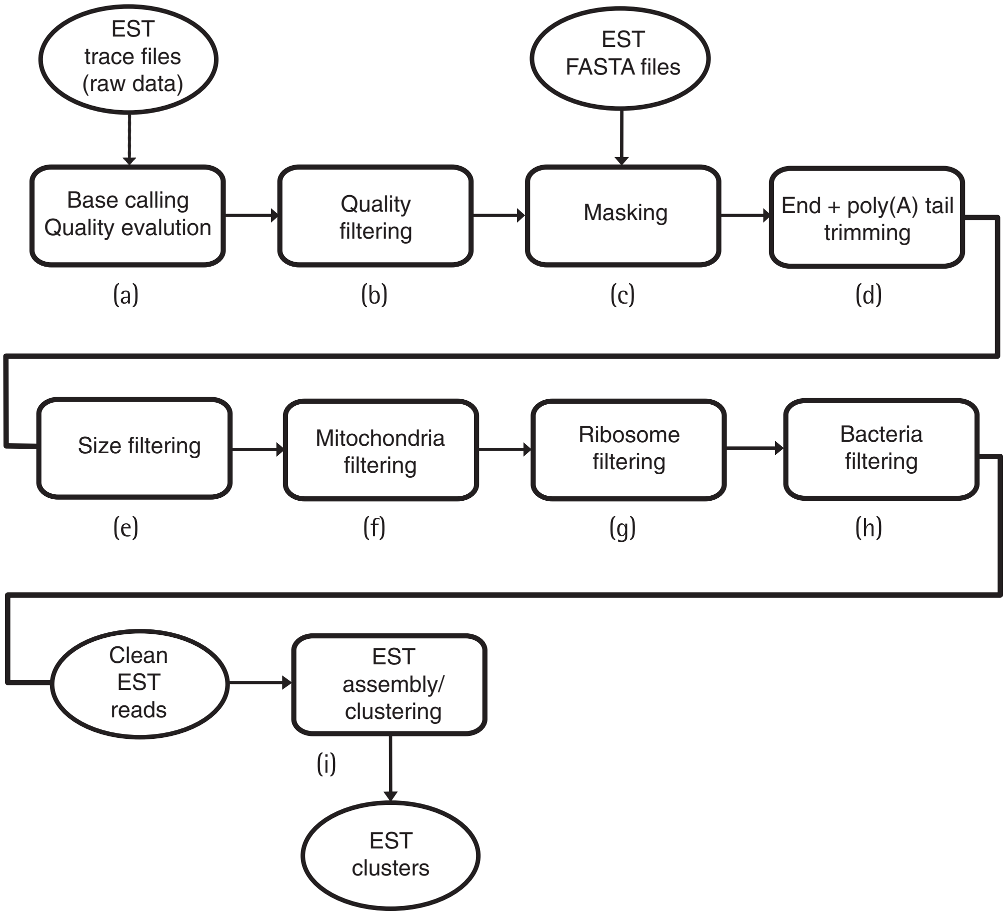


Figure 3. A typical EST processing pipeline.
An EST processing pipeline is composed of several sequential processing steps in which the output data from one step is used as the input for the subsequent one. The scheme shows a typical pipeline. Input data can be loaded into the pipeline using either trace files or FASTA format sequence files. The chromatogram files are processed by PHRED (34, 35), a program that performs base calling and quality evaluation for each base (*a*). The reads are submitted to a second component (*b*) that filters out the reads that have an overall quality below a user-defined threshold. (If FASTA sequence files are used instead of trace files, no quality evaluation can be performed and steps *a* and *b* are skipped.) Sequences then are processed by a component (*c*) that runs CROSS_MATCH (38) to mask sequences arising from the cloning vector or from the primers used to generate the ESTs. Sequences are then submitted to a trimming step where low-quality/masked regions and poly(A) tails are trimmed off (*d*), and the reads whose remaining sequences after trimming fall below a user-defined minimum length are discarded by a size filter (*e*). Next, sequences are processed by multiple pipeline components that run either CROSS_MATCH (*f*, *g*) or BLAST (*h*, *i*) to perform similarity searches against databases of undesired sequences and potential contaminants, which include mitochondrial (*f*), ribosomal (*g*), and bacterial (*h*) sequences; all reads presenting alignment blocks above a set of user-defined values are filtered out. Finally, the ESTs are assembled (*i*) using CAP3 (28). The clean EST assembled clusters are then ready for annotation.

The details of the EST pipeline will, of course, be tailored to suit the dataset being analyzed. *Protocol 4* describes the construction of a typical EST processing pipeline using EGENE, a generic pipeline generation system (39). The pipeline will be used for processing an example dataset consisting of trace files of a cDNA library synthesized through the ORESTES method (12). The suggested parameters of each step are known to work well for this dataset, but should also work reasonably well for most EST sequencing projects. The reader can use this example pipeline as a template and then change the number and order of the processing steps, as well as the corresponding databases, in order to fit specific project requirements. Replacing databases and inserting or removing pipeline components can be performed easily with COED, EGENE's graphical configuration editor (see *Fig. 4*). For more details on how to use EGENE and COED, the reader is referred to the original description (39) and the official web site (see below). All programs or symbolic links should be put on a directory that is specified on the path of the operating system. Should you get any installation problems, please refer to the respective official pages and publications of the software.

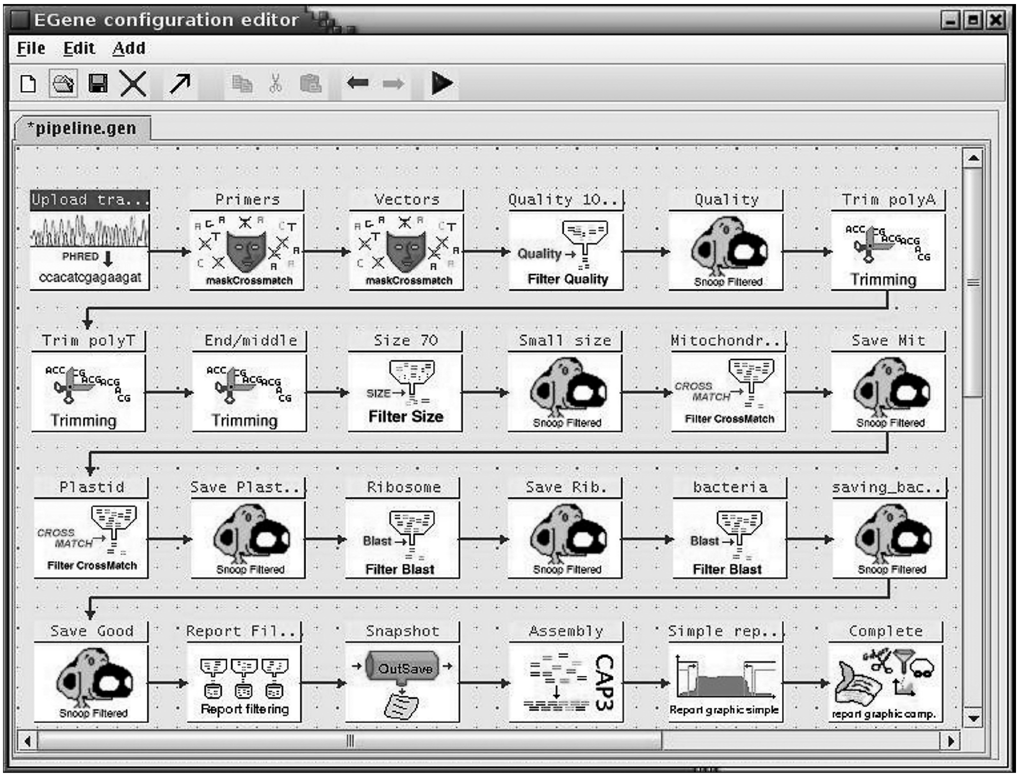


Figure 4. Building pipelines with EGENE and COED.
EGENE is a generic, flexible, and modular pipeline generation system that makes pipeline construction a modular job. Pipelines can be constructed using COED, a Java visual configuration editor. Icons representing each component of the pipeline are selected and landed on the canvas. The different steps are interconnected using an arrow tool and the pipeline can be executed within COED (clicking on the arrowhead button) or directly from the UNIX command line.

The example pipeline to be used in this tutorial will be run on a set of chromatogram files of ORESTES reads of *Eimeria tenella*, an apicomplexan protozoan parasite. All reads will be submitted to contaminant filtering steps against mitochondrial, apicoplast (an ancient plastid-derived organelle), ribosomal, and bacterial sequences. In detail, the pipeline consists of the following steps:

1. Uploading trace files and performing base calling and quality evaluation.
2. Masking primer sequences.
3. Masking vector sequences.
4. Filtering low-quality sequences.
5. Saving sequences invalidated by the quality filter.
6. Trimming off poly(A) and…
7. …the complementary poly(T)) sequences.
8. Trimming off bases that present a low PHRED quality value and those that are masked.
9. Filtering short sequences.
10. Saving sequences invalidated by the size filter.
11. Filtering mitochondrial sequences.
12. Saving sequences invalidated by the mitochondrial contaminant filter.
13. Filtering plastid sequences.
14. Saving sequences invalidated by the plastid contaminant filter.
15. Filtering ribosomal sequences.
16. Saving sequences invalidated by the ribosomal contaminant filter.
17. Filtering bacterial sequences.
18. Saving sequences invalidated by the bacterial contaminant filter.
19. Saving sequences not previously invalidated by any filter.
20. Generating a report of all filtering steps.
21. Creating an XML snapshot recording all of the processing steps that were performed.
22. Assembling the valid sequences using CAP3.
23. Generating an HTML page with graphical reports.
24. Generating a complete graphical report.

Protocol 4

Building an automated pipeline for EST processing

Software

You will need the following software:

- EGENE (38) (http://www.coccidia.icb.usp.br/egene/ 7.10)
- PHRED, PHRAP, CONSED, and CROSS_MATCH (34, 35, 39, 40). These programs should be requested from the authors. Contact information and instructions are available at http://www.phrap.org 7.11
- BLAST (41) (http://www.ncbi.nlm.nih.gov/BLAST/ 7.12)

- CAP3 (28) (http://seq.cs.iastate.edu/ 7.13)
- Perl interpreter, version 5.6.0 or higher (http://www.perl.org 7.1), with the GD graphics library (http://www.boutell.com/gd/ 7.14), DBD::Pg
- Java 2 Platform, Standard Edition (J2SE), version 1.4.1 or higher (http://java.sun.com/j2se/ 7.2)

Method

1. We have previously constructed a pipeline for this tutorial using COED and saved the configuration file as pipeline.gen 7.15 in the /Protocol_4/config_files directory. In order to run the pipeline, go to the /Protocol_4/dataset directory. This directory contains the chromat_dir subdirectory, which presents a set of 96 trace files.
2. Invoke COED on the UNIX/Linux command line by typing '`coed.pl`' and pressing 'enter'.
3. Using the menu bar, choose the **File Open** command and select the pipeline.gen file, located in the /Protocol_4/config_files directory.
4. You should see a graphical display of the pipeline (see *Fig. 4*) with all components represented as interconnected icons.
5. Any component can be deleted simply by selecting it with the mouse and using the **Edit Cut** command. Please refer to the EGENE web site for a complete list of commands.
6. To run the pipeline within COED, click on the green arrowhead button of the toolbar.
7. COED will open up a dialog box asking for the working directory. Click on the blank form with the right button of the mouse and select the directory /Protocol_4/dataset (a full directory path must be specified, according to the directory structure of your local server). Press the **Run** button.
8. COED will now display a small window informing you that your pipeline is executing. Because the multiple processing steps may take a long time to run, the pipeline will be executed in the background. Thus, COED will not report the end of the entire job and can now be closed.
9. To check whether the pipeline processing has finished, you have to use the operating system's command line. Type '`ls -l`' to list the files. No temporary directories (identified by a '`_temp_`' suffix) should be present when the pipeline processing is finished. The pipeline used in this tutorial processes 96 reads and takes around 2 min to run in a PC/Linux with a Pentium 4 2.8 GHz processor.
10. Alternatively, instead of running the pipeline from within COED, you can also save your pipeline as a *.cnf text file using the **Save As EGene file** command.
11. Assuming that the pipeline.cnf configuration file has been stored in the /Protocol_4/config_files directory, and that you are in the /Protocol_4/dataset directory, the UNIX/Linux command to execute the pipeline is:

```
bigou.pl -c ../config_files/pipeline.cnf >/dev/null&
```

This command invokes `bigou.pl`, an EGENE program that reads the configuration files and starts each processing step of the pipeline. Although it is not necessary, we recommend that you redirect the standard output to a null device (specified by a '`>/dev/null`' statement at the end of the command) in order not to get your screen jammed with messages. Also, using the '`&`' character at the end of the command will put the whole process in the background, thus liberating the terminal. At the end of the process, you should find the following additional files in this directory:

filtered_by_quality.fasta
filtered_by_size.fasta
filtered_by_mitochondrion.fasta

filtered_by_plastid.fasta
filtered_by_ribosome.fasta
filetered_by_bacteria.fasta
filtering_report.html
redundancy_report.html
report_graphic_simple.html
final_snapshot.xml
good_sequences.fasta

and also the additional subdirectories:

assembly_dir
complete_report
images_dir

12. Check the content of the results files containing the prefix 'filtered_by_' by typing '`more name_of_the_file`' or loading them onto any text editor. These files contain multiple sequences in a FASTA format and correspond to the reads filtered out in each of the filtering steps. If no read is identified by a filtering step, then the corresponding file will be empty.
13. The file named good_sequences.fasta contains all sequences that passed through the filtering steps and were considered 'good'. These sequences have also fulfilled the minimum quality criteria established on the pipeline. Please note that the sequences are trimmed, so they do not contain any vector, primer, or low-quality bases.
14. The assembly_dir directory contains a typical directory structure required by CONSED to visualize DNA assemblies. Thus, chromat_dir contains the trace files, phd_dir stores the PHD (PHRED-processed) files of the accepted sequence reads, and edit_dir contains the assembly files created by the CAP3 assembler. To inspect the DNA assembly, invoke CONSED within the edit_dir directory. The command varies according to the CONSED version you are using but, in general, it should be '`consed`' or '`consed_linux`':

    ```
    consed clean.fasta.cap.ace
    ```

    Once CONSED is loaded, select the clean.fasta.cap.ace file. All contigs will be listed. Choose one of them by clicking twice with your mouse. A new window will open up displaying the multiple sequence assembly view. For more information on how to use CONSED, please consult the original documentation of the software.
15. A complete set of the expected results is provided in the /Protocol_4/results directory.

2.4 Transcript reconstruction

One can trace a parallel between an unclustered set of EST reads and a bulk of construction bricks. In the same way that bricks turn out to be much more useful when arranged with each other in a shape of a house, ESTs become much more informative when adequately clustered and assembled in a process known as transcript reconstruction. In fact, EST data has a fragmentary character and presents the following limitations:

- **Short length.** ESTs vary from 100 to 700 bp, with a typical average of 400 bp (see *Fig. 1*). In most cases, this size is not enough to cover whole transcripts.
- **Low-quality data.** ESTs are single-pass reads and only a relatively small portion of the sequence presents a high quality and good confidence level.

- **Nonoverlapping reads may cover the same transcript.** ESTs may cover the same transcript on different nonoverlapping regions, leaving sequence gaps.
- **EST libraries represent subsets of the transcriptome.** EST libraries only reflect the gene expression profile of the cell/tissue used as the source of mRNA, thus representing only a fraction of the transcriptome of the whole organism.
- **Representational bias.** Due to the heterogeneous frequency of transcripts, even a large sampling may still miss some rare transcripts. Biases of cDNA synthesis and cloning contribute towards making this representation worse.

To reduce data fragmentation and extract all potential information, ESTs must be submitted to a process known as transcript reconstruction. It is beyond the scope of this chapter to cover in much detail the different approaches proposed for such a task. For in-depth descriptions, the reader is referred to reviews (5, 42, 43) and the original articles describing the protocols utilized for STACKdb (29-31), TGI (26, 44, 45), and UniGene (14, 46). Here, we will instead discuss general concepts involved with this issue and then introduce the most important EST resources.

EST clustering and assembly are often and erroneously used synonymously. Clustering can be defined as the process of grouping subsets of EST reads that share some sequence among themselves. Thus, clustering involves an all-versus-all comparison using a loose stringency and, preferably, fast algorithms. Once the clusters are established, then a DNA assembler can be used to align the overlapping reads of each cluster and generate consensus sequences. For small sets of ESTs, a single assembly step can be used without previous clustering (6). Bypassing the clustering step, however, may present potential disadvantages. First, because specific clustering programs do not perform a real pairwise sequence alignment, but rather look for sequence words shared by distinct reads (31, 47), they are often much faster than conventional DNA assembly software. This is particularly more pronounced in large EST sets, where a direct assembly process is not always feasible. Secondly, clustering word-sharing sequences makes more sense than going straight to the assembly phase, as sequences representing different paralogous genes or splicing variants can be clustered together and then, after a proper assembly, can be separated into different contigs. This approach preserves the information about which consensus sequences belong to the same cluster and, as such, are closely related. This method (see *Fig. 5*) is employed by STACKPACK, one of the most complete transcript reconstruction solutions (30, 31, 43). TGI (26, 44, 45, 48), on the other hand, uses a protocol that consists in an all-versus-all sequence comparison using a modified version of MEGABLAST (27). Transitive closure is used for building the clusters using a criterion where sequences presenting overlaps of at least 40 bp and 95% identity, with a maximum mismatched overhang of 30 bp, are included in the same cluster. Tentative consensus sequences are then generated for each cluster using the CAP3 assembler (28).

Clustering can be performed using either loose or stringent conditions. A loose clustering may result in paralogs being clustered and assembled together, but consensus sequences tend to be longer. Conversely, by using a stringent clustering, one can differentiate paralogs more accurately, but the consensus sequence length of each cluster will be rather shorter. There is no universal recipe

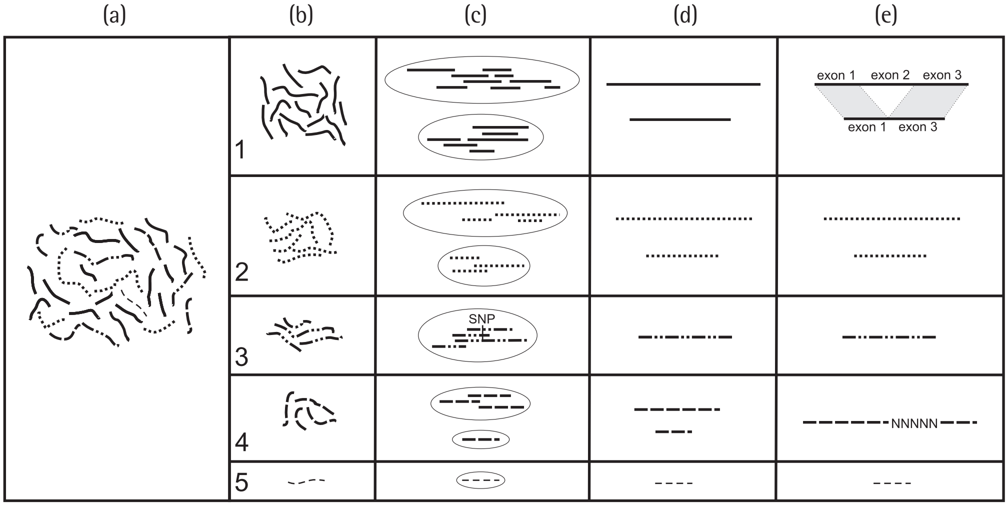


Figure 5. EST clustering and analysis.
Transcript reconstruction is a complex multi-step task. The scheme is based on the STACKPACK (31, 42) pipeline and shows the different steps involved in a process aimed at extracting all potential EST information. The columns of the chart represent the different phases of an EST clustering and analysis process. A set of ESTs (*a*) is submitted to a clustering step (*b*), resulting in clusters (numbered 1–5). Each cluster may be composed of multiple reads, or may consist of a single read (a 'singleton', such as cluster 5). Each cluster is assembled separately (*c*), to generate either a single contig (clusters 3 and 4) or multiple contigs (clusters 1 and 2). Some reads may fail to assemble into contigs, remaining as singletons (as is the case for one read of cluster 4). At this stage, candidate SNPs can be identified in the multiple sequence alignment (*c*, cluster 3). Consensus sequences are derived from each assembly (*d*) and if clone information is available, 5′ and 3′ reads of the same clones can be linked (cluster 4, column *e*). Also, contigs can be aligned with each other for the identification of alternate splicing forms (cluster 1, column *e*).

for establishing a high-confidence clustering without some manual curation, a step needed for checking whether the cluster separation follows a biological sense. The appropriate stringency is very difficult to define *a priori*, as different gene families may present distinct divergences across their respective paralogs due to the different paralogy times. Furthermore, different substitution rates can be observed among gene families. As a result, a gene family may contain paralogs that are much more closely related to each other than those of another family are among themselves.

Following the clustering and assembly steps, a set of clusters is obtained, each comprising one or more contigs composed of multiple reads. The process may also result in some clusters or contigs that consist of only a single read – 'singletons' (see *Fig. 5*). Singletons may correspond to low-expression transcripts that were collected only once in the EST sampling. Alternatively, they may have been originated from an unidentified contaminant source, so a manual inspection must be performed before considering singletons as rare transcript representatives. If unidirectional libraries have been used, and information on sequence direction is available, then it is possible to perform clone linking (see *Fig. 5*). This is analogous to the use of 'read pairs' in shotgun sequence assembly: two clusters can be linked if one contains a sequence read from the 5′ end of an insert and the other contains a sequence read from the 3′ end of the same clone (43). Finally, a deeper inspection may reveal splicing variants and SNPs. We will not cover here methods for SNP surveying, and the reader is advised to consult specific references (49–52).

The following protocol demonstrates the use of the TGI Clustering Tools package (TGICL) and CLVIEW, a graphical interactive tool for visualization of ACE format assembly files generated by CAP3 or PHRAP. TGICL and CLVIEW were developed at TIGR and are freely available for download at the web address below. We will use the *P. falciparum* EST dataset downloaded in *Protocol 2*.

Protocol 5

Clustering ESTs using TGICL

Software

You will need the following software:

- TGICL and CLVIEW (53) (http://compbio.dfci.harvard.edu/tgi/software/[7.16])

Method

1. Go to the directory where you stored the *P. falciparum* EST dataset that you downloaded from dbEST in *Protocol 2*. Alternatively, go to the /Protocol_5/dataset directory, where you will find P_falciparum.fasta[7.17] (this is identical to the file provided in the Protocol_2 folder, and is a set of *P. falciparum* ESTs downloaded in October 2006). The analysis that follows uses this stored dataset, and the results will differ slightly if you use a more recently downloaded set of ESTs.

2. TGICL is very simple to run in a default mode. Type the following command:

```
tgicl P_falciparum.fasta
```

This process can take several minutes. When finished, you should receive the following message on the screen:

```
tgicl (P_falciparum.fasta) finished on machine in /your_ ↵
   directory/P_falciparum, without a detectable error
```

3. Using UNIX's '`ls`' and '`less`' commands, identify all files and directories that were created. (A complete set of the results from the dataset provided is given in the /Protocol_5/results directory.)

4. We can now answer the following questions:

(i) How many clusters were generated?
TGICL creates a file called *_clusters (where * is the name of the input file), which presents a pseudo-FASTA format where each record is actually a cluster definition and consists of a header line containing a greater than ('>') character. Thus, to count the number of clusters, we just have to use the command:

```
grep ">" P_falciparum.fasta_clusters | wc -l
```

If you used the dataset provided (P_falciparum.fasta), you should find that there were 2782 clusters.

(ii) How many singletons were generated?
Singletons are sequences that are not clustered with others – they represent 'single-

sequence clusters'. TGICL stores a list of all singleton reads in a file called file_name.singletons. If you used the dataset EST.fasta, the file should be P_falciparum.fasta.singletons. As the sequence headers from NCBI use a '`gi`' (gene identifier) tag, we can count the number of reads simply by counting how many times that the string '`gi`' appears in the file:

```
grep "gi" P_falciparum.fasta.singletons | wc -l
```

Using the dataset provided, you should find that there were 5297 singletons.

(iii) How can one retrieve the singleton sequences?
TGICL creates a *.cidx index file for fast retrieval of cluster sequences. By using the CDBYANK program (provided in the TGICL package), we can extract the singleton sequences from the index file and store them in a file named singleton.seqs using the command:

```
cdbyank P_falciparum.cidx < P_falciparum.fasta.singletons ↩
   > singleton.seqs
```

(iv) How many contigs were generated?
TGICL creates an asm_X subdirectory for each central processing unit (CPU) in a parallel processing setting (where *X* is a number ≥1) or just one if a single CPU is used. In our case, all assembly files will be stored within the asm_1 subdirectory. All contig sequences are stored in a multiple-sequence FASTA file called contigs. To count the number of sequences, type the following command within the /asm_1 directory:

```
grep ">" contigs | wc -l
```

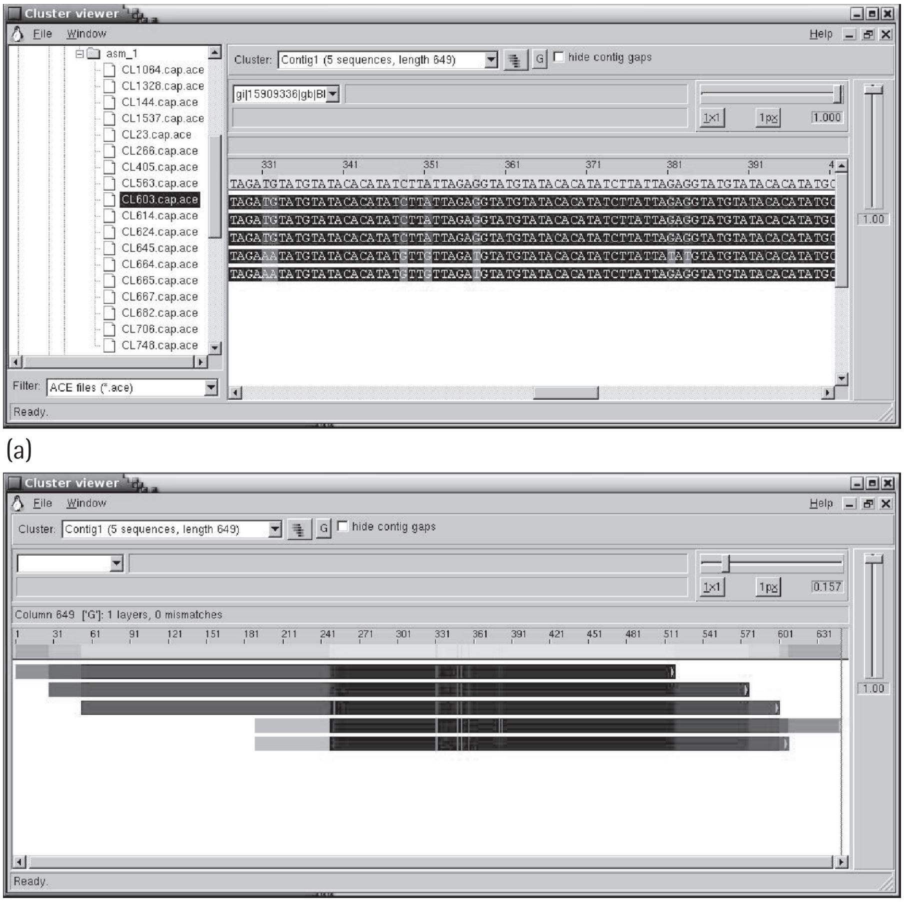
If you used the dataset provided, you should find that there were 3166 contigs.

(v) How many singlets were generated?
Singlets (as distinct from singletons) are sequences that are initially clustered with others by sharing some level of similarity. However, this similarity is not high enough to allow them to be assembled with any other read into a contig. Singlet sequences are stored in a multiple sequence FASTA file called singlets. To count the number of singlets, just type:

```
grep ">" singlets | wc -l
```

If you used the dataset provided, there should be 133 singlets.

5. We can visualize some contigs using the CLVIEW tool, a graphical program that allows viewing of the ace assembly files. Invoke CLVIEW by typing '`clview`' on the command line.
6. A directory tree will be displayed on the left section of the window. Select **All Files** on the **Filter** selection box located at the left bottom corner. Now, using the directory tree, select the ace file stored within the /asm_1 directory and load it.
7. The window (see *Fig. 6a*, also available in the color section) will now present, on the right, a section displaying a consensus sequence (with a yellow background) followed by a stack of aligned sequences (with a blue background). The blue background is in different shades, which indicate the sequence coverage: darker shades represent regions matched by a high number of reads, whilst paler shades are regions matched by fewer reads.
8. The upper part of the right window presents a selection box (labeled **Cluster:**) that allows you to select any cluster for visualization. In the upper-right corner, there are two bars for zooming the assembly (horizontal bar) and to scroll the reads (vertical bar). *Fig. 6*(*b*) shows the same cluster with a wider (zoomed-out) view.
9. Base discrepancies are represented by red bases (see *Fig. 6a*) or vertical bars (see *Fig. 6b*), depending on the zoom used for visualization. These discrepancies may be due to sequencing errors, but can also represent potential SNPs (see section 3).



Cluster: Contig1 (5 sequences, length 649)

□ hide contig gaps

gi|15909336|gb|BI

1.000

Filter: ACE files (*.ace)

```
 331       341       351       361       371       381       391       4
TAGATGTATGTATACACATATCTTATTAGAGGTATGTATACACATATCTTATTAGAGGTATGTATACACATATGC
TAGATGTATGTATACACATATCTTATTAGAGGTATGTATACACATATCTTATTAGAGGTATGTATACACATATGC
TAGATGTATGTATACACATATCTTATTAGAGGTATGTATACACATATCTTATTAGAGGTATGTATACACATATGC
TAGATGTATGTATACACATATCTTATTAGAGGTATGTATACACATATCTTATTAGAGGTATGTATACACATATGC
TAGAAATATGTATACACATATGTTGTTAGATGTATGTATACACATATCTTATTATATGTATGTATACACATATGC
TAGAAATATGTATACACATATGTTGTTAGATGTATGTATACACATATCTTATTAGAGGTATGTATACACATATGC
```

Ready.

(a)

Cluster: Contig1 (5 sequences, length 649)

□ hide contig gaps

0.157

Column 649 ['G']: 1 layers, 0 mismatches

Ready.

(b)

Figure 6. Visualizing cluster assemblies using CLVIEW (see page xxii for color version). CLVIEW presents cluster assemblies in a zoomed view (*a*), displaying a directory tree on the left of the window and the aligned DNA sequences on the right; or in an overview of the assembled reads (*b*), displaying the consensus sequence with a yellow background, followed by a pile of aligned reads marked with a blue background. Base discrepancies are labeled in red and may represent potential SNPs.

2.5 Redundancy estimation

Redundancy assessment of EST libraries generally starts by using assembly programs such as CAP3 (28) or PHRAP (38), or a clustering step followed by assembly. An initial analysis may involve an estimation of the proportion of clusters in regard to the total number of reads of an EST set. By subtracting this value from unity, one can estimate the overall read redundancy. This value, however, can be misleading, as it does not tell us precisely how much information is in fact being generated. Hence, a hypothetical situation where all reads span only a single small region of a transcript would yield the same redundancy as if these reads were covering the entire sequence of this transcript. In the latter case, the overall gain of information would certainly be higher that in the former. A possible solution to overcome this limitation is to calculate the base redundancy rather than the read redundancy. In this case, the total number of bases of the consensus sequences would be divided by the sum of bases of the separate reads, thus allowing a measurement of how much novel information has been gained following the incorporation of new reads. If the addition of new reads keeps revealing more transcripts and/or covering novel transcript regions (and hence more bases in total), one should expect base redundancy to be lower than read redundancy. On the other hand, once maximum coverage has been attained, one should expect base and read redundancy to tend towards equity.

Another aspect that has to be taken into account is the complexity of the transcriptome (the total set of transcribed genes). For instance, 1 million reads from a highly representative and unbiased EST library would probably not result in a high redundancy for the human transcriptome (estimated at 30 000 to 100 000 transcripts). Conversely, a similar set of reads would yield a much higher redundancy in a small transcriptome (e.g. 5000 transcripts). *Fig. 7* shows the typical progress of the cluster number as new EST reads are added. At the very early stages of EST sequencing, every single read corresponds to a new cluster. The number of clusters at this stage keeps growing linearly until a point is reached where a new read is as likely to connect two clusters as to form a new cluster and the curve levels off. From this turning point, cluster joining outweighs cluster formation and the number of clusters declines until a final situation is reached in which every cluster corresponds roughly to a reconstructed transcript. However, in real life, there will always remain some gaps in the reconstructed transcripts (inflating the number of clusters), whilst some rare transcripts may not be represented at all (reducing the number of clusters) – the balance between these effects will lead to an eventual over- or underestimate of the total transcriptome size. Finally, it is also important

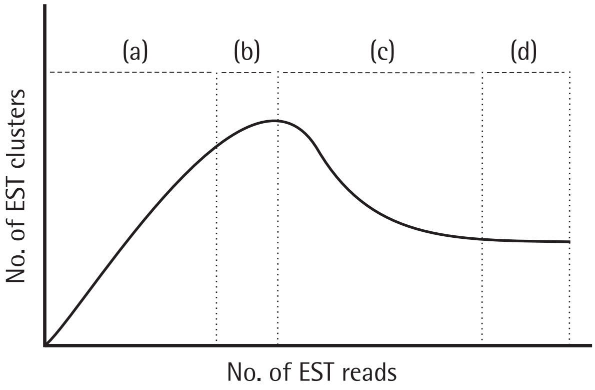


Figure 7. Progress of the cluster number in a typical EST sequencing project.
In the early stages of an EST sequencing project (*a*), most of the reads do not overlap one another and so each new EST initiates a new cluster, resulting in a linear increase in the number of clusters. As more reads are accumulated (*b*), overlaps are found and cluster joining begins to offset the initiation of new clusters. Later (*c*), cluster joining outweighs initiation and the total number of clusters decreases towards an eventual plateau (*d*), where all new ESTs fall into existing clusters.

to take into account the fact that splicing variants can make this scenario even more complex.

2.6 Electronic gene expression profiles

The relative abundance of EST reads may roughly reflect the gene expression profile of the respective tissue/cell type used as the source of mRNA. This assumption must be considered with caution, as several factors can interfere with the quantification. First, the relative abundance of EST reads only reflects the gene expression profile if the EST library has not been normalized. Any significant normalization, as well as construction and cloning biases, may hamper quantitative analyses, as they change the original distribution of the different classes of transcript. Secondly, correct clustering is important for the accuracy of electronic gene expression analysis. For instance, if a read is erroneously grouped to a certain cluster, it will contribute to alter the quantification of the corresponding transcript. Finally, it is worth mentioning that transcript sampling may introduce errors, and comparative expression analysis using different libraries has to take into account systematic biases introduced during cDNA generation (54).

2.7 Mapping ESTs to the genome

Gene structure is much more complex in eukaryotes than in prokaryotes. The discontinuous character of genes in the former, characterized by the presence of coding regions intercalated by intronic noncoding intervening sequences, makes gene prediction a difficult task, especially for identifying small exons. This can be still more complicated by the occurrence of alternative splicing events such as exon skipping, in which an exon may be present in one transcript but be skipped in a splicing variant. The fast-growing nature of EST databases makes them an invaluable tool for gene structure determination. Alignment of cDNAs with genomic sequences may provide important evidence to support gene predictions, but is also not a trivial task. The intervening nature of the introns breaks up the sequence alignments and small exons may remain undetected. Furthermore, most pairwise alignment programs such as FASTA (55) and BLAST (41) do not implement any modeling of the exon–intron structure and the presence of canonical acceptor and donor splicing sites. For this reason, such programs may generate alignment blocks whose ends do not correspond to the exact intron–exon boundaries (56). Alternative alignment programs that can handle introns and are used for cDNA mapping include EST2GENOME (57), SPIDEY (58), BLAT (59), SIM4 (53), and EXONERATE (60). Nevertheless, whatever the program utilized for transcript mapping, the appropriate alignment stringency has to be determined case by case, especially if cross-species cDNA mapping is performed.

In the next protocol, we will see how to map cDNA sequences onto genomic sequence using EXONERATE (60), a flexible and configurable program that allows rapid searches thanks to the use of heuristics based on alignment models. For this purpose, we will use three human cDNA sequences: hemoglobin β-chain

(GenBank accession no. BC007075.1), glyceraldehyde 3-phosphate dehydrogenase (NM_002046.3) and calmodulin 1 (NM_006888.3). These sequences will be mapped onto stretches of the human chromosomes 11, 12, and 14, respectively. To map cDNAs onto genomic sequences, we will use the EST2GENOME model of EXONERATE, which includes intron modeling and permits alignment of spliced transcript sequences to the unspliced genomic sequence. (Chapter 4 also covers the mapping of ESTs to genomic sequence, using a web server.)

Protocol 6

Mapping cDNAs onto genomic sequences

Software

You will need the following software:

- EXONERATE (60) (http://www.ebi.ac.uk/~guy/exonerate/[7.18])

Method

1. Go to the /Protocol_6/dataset directory where you will find two files: cDNAs.fasta[7.19] and genomic.fasta[7.20]. These files contain three cDNA sequences and three genomic sequences, respectively, in FASTA format.
2. EXONERATE can align two single- or multiple-sequence files against one another. The sequences are aligned all-versus-all and the relevant alignments stored in a single output file. To align the three cDNA sequences with the respective chromosomal sequences using EXONERATE, type the following command (as a single line):

```
exonerate cDNAs.fasta genomic.fasta --model est2genome ↵
   --score 300 --showtargetgff > cDNA_X_genomic_map
```

   This command invokes the program EXONERATE, specifies the names of the query and subject sequence files, respectively, and redirects the output ('>' sign) to a specified file name, cDNA_X_genomic.txt. The parameter '--model est2genome' specifies intron modeling. A value of 300 (default value is 100), specified in the '--score' parameter, corresponds to a moderate stringency. The parameter '--showtargetgff' generates a generic file format (GFF) output following the alignment and is convenient for downstream annotation.

3. Inspect the content of the newly created file (cDNA_X_genomic_map) using any text editor or the UNIX 'less' command. It contains the corresponding cDNA-to-genome alignments. EXONERATE displays the sequence alignment blocks with a clear indication of the introns found. (A copy of the expected results file is given in the /Protocol_6/results directory.)

3. TROUBLESHOOTING

3.1 Clone chimerism

The typical short length of EST clones encourages insert-to-insert ligation and cloning during library construction. Chimeric clones, characterized by the concomitant cloning of two fragments derived from different genes, are a common finding in EST libraries (61). Detection of such artifacts is a complex task and may require a manual inspection of the sequences. Furthermore, chimerism must be interpreted with caution and discriminated from chimeric spliced mRNAs (62, 63).

3.2 SNPs

SNPs observed in different EST reads must be interpreted with caution, as they may be a consequence of sequencing errors rather than a natural occurrence of polymorphic sites. Read redundancy and base quality must be checked on each candidate polymorphic site to discard potential artifacts.

3.3 Repeat masking

Masking low-complexity sequences such as tandem repeats before clustering and assembling ESTs may improve the accuracy of the process. However, aggressive masking may prevent the correct grouping of legitimate clusters. On the other hand, lack of masking may result in the formation of chimeric clusters, composed of unrelated sequences that share some repetitive sequences (15).

3.4 Contamination

EST processing pipelines can incorporate filters against heterologous and organellar sequences. Conversely, cross-contamination of mRNA sources with surrounding tissues, other cell types, or different developmental stages can barely be discriminated by conventional software. Hence, any assertion regarding expression specificity must be corroborated by specific experimental approaches.

4. ADDITIONAL WEB RESOURCES

- A Science Primer: http://www.ncbi.nlm.nih.gov/About/primer/index.html[7.21]. A series of introductory texts produced by the National Center for Biotechnology Information (NCBI) that cover topics such as bioinformatics, ESTs, SNPs, and genome mapping, among others.
- Submitting sequences to dbEST and GenBank: http://www.ncbi.nlm.nih.gov/dbEST/how_to_submit.html[7.22]. EST submission must follow a standardized format. This page provides a guideline for EST submission to dbEST.
- The GenBank Expression Sequence Tags Database (dbEST) (23, 24): http://www.ncbi.nlm.nih.gov/dbEST/[7.23]. A public repository with tens of millions of EST entries.
- UniGene (14, 46): http://www.ncbi.nlm.nih.gov/UniGene/[7.9]. An integrated database of clustered ESTs and cDNA sequences. Each UniGene cluster presents accession numbers to ESTs, IMAGE clones, etc., plus some informative annotations. No consensus sequences are provided.

- TIGR Gene Indices (26, 44): http://compbio.dfci.harvard.edu/tgi/ 7.7. A resource of species-specific EST data of eukaryotic organisms, including animal, plant, protist, and fungal sources. ESTs are clustered, assembled, and annotated. The data is stored in a relational database with a user-friendly web interface. Tentative consensus sequences are provided.
- The Sequence Tag Alignment and Consensus Knowledgebase (STACKdb) (26, 29, 44): http://www.sanbi.ac.za/Dbases.html 7.8. A database of clustered human EST and mRNA sequences, including some disease- and tissue-based categories, and variation analysis.
- Genome Sequencing Center at the Washington University in Seattle: http://genome.wustl.edu/data/est.cgi 7.24. A large amount of sequencing data is available on this site, including both genome and EST projects of many organisms. EST trace files are available for download at ftp://genome.wustl.edu/pub/est/ 7.25.
- NCBI Reference Sequence (RefSeq) (64): http://www.ncbi.nlm.nih.gov/RefSeq/ 7.26. A comprehensive, integrated, nonredundant set of sequences for major research organisms, including genomic DNA, transcripts (RNA), and protein products.
- NCBI Trace Archive: http://www.ncbi.nlm.nih.gov/Traces/ 7.27. A public repository with more than one billion trace files. Data is exchanged regularly with the Ensembl Trace Server (http://trace.ensembl.org/ 7.28) at the EBI/Sanger Institute in the UK.
- NCBI dbSNP (65): http://www.ncbi.nlm.nih.gov/SNP/ 7.29. A database of nucleotide sequence variation that includes SNPs, deletion/insertion polymorphisms, microsatellite or short tandem repeats, and multi-nucleotide polymorphisms.

5. REFERENCES

★ 1. **AdamsMD, Kelley JM, Gocayne JD, *et al.*** (1991) *Science*, **252**, 1651–1656. – *The original publication describing the concept and use of ESTs.*

2. **Adams MD, Dubnick M, Kerlavage AR, *et al.*** (1992) *Nature*, **355**, 632–634.

3. **Adams MD, Kerlavage AR, Fields C & Venter JC** (1993) *Nat. Genet.* **4**, 256–267.

4. **Gill RW & Sanseau P** (2000) *Biotechnol. Annu. Rev.* **5**, 25–44.

★★ 5. **JongeneelCV** (2000) *Brief. Bioinform.* **1**, 76–92. – *A good overview on EST processing and annotation.*

6. **Lindlof A** (2003) *Appl. Bioinform.* **2**, 123–129.

7. **Marra MA, Hillier L & Waterston RH** (1998) *Trends Genet.* **14**, 4–7.

★ 8. **ParkinsonJ & Blaxter M** (2004) *Methods Mol. Biol.* **270**, 93–126. – *A theoretical and practical text depicting the most important steps in EST processing using* CLOBB *software.*

9. **Ying SY** (2004) *Mol. Biotechnol.* **27**, 245–252.

10. **Ying SY** (2003) *Generation of cDNA Libraries – Methods and Protocols*. Humana Press Inc., Totowa, NJ.

11. **Dias-Neto E, Correa RG, Verjovski-Almeida S, *et al.*** (2000) *Proc. Natl. Acad. Sci. U.S.A.* **97**, 3491–3496.

12. **Dias-Neto E, Harrop R, Correa-Oliveira R, Wilson RA, Pena SD & Simpson AJ** (1997) *Gene*, **186**, 135–142.

13. **Adams MD, Soares MB, Kerlavage AR, Fields C & Venter JC** (1993) *Nat. Genet.* **4**, 373–380.

★ 14. **BoguskiMS & Schuler GD** (1995) *Nat. Genet.* **10**, 369–371. – *The original publication describing UniGene.*

15. **Jongeneel CV** (2000) *Bioinformatics*, **16**, 1059–1061.

16. **Wilcox AS, Khan AS, Hopkins JA & Sikela JM** (1991) *Nucleic Acids Res.* **19**, 1837–1843.

17. **Sikela JM & Auffray C** (1993) *Nat. Genet.* **3**, 189–191.

18. **Soares MB, Bonaldo MF, Jelene P, Su L, Lawton L & Efstratiadis A** (1994) *Proc. Natl. Acad. Sci. U.S.A.* **91**, 9228–9232.

19. **Bonaldo MF, Lennon G & Soares MB** (1996) *Genome Res*, **6**, 791–806.

20. **Patanjali SR, Parimoo S & Weissman SM** (1991) *Proc. Natl. Acad. Sci. U.S.A.* **88**, 1943–1947.

21. **Diatchenko L, Lau YF, Campbell AP, *et al.*** (1996) *Proc. Natl. Acad. Sci. U.S.A.* **93**, 6025–6030.

22. **Hedrick SM, Cohen DI, Nielsen EA & Davis MM** (1984) *Nature*, **308**, 149–153.

★ 23. **BoguskiMS, Lowe TM & Tolstoshev CM** (1993) *Nat. Genet.* **4**, 332–333. – *The original publication describing dbEST.*

24. **Boguski MS, Tolstoshev CM & Bassett DE Jr** (1994) *Science*, **265**, 1993–1994.

25. **Benson DA, Karsch-Mizrachi I, Lipman DJ, Ostell J & Wheeler DL** (2005) *Nucleic Acids Res* **33**, D34–D38.

26. **Quackenbush J, Liang F, Holt I, Pertea G & Upton J** (2000) *Nucleic Acids Res.* **28**, 141–145.

27. **Zhang Z, Schwartz S, Wagner L & Miller W** (2000) *J. Comput. Biol.* **7**, 203–214.

28. **Huang X & Madan A** (1999) *Genome Res.* **9**, 868–877.

29. **Christoffels A, van Gelder A, Greyling G, Miller R, Hide T & Hide W** (2001) *Nucleic Acids Res.* **29**, 234–238.

30. **Miller RT, Christoffels AG, Gopalakrishnan C, *et al.*** (1999) *Genome Res.* **9**, 1143–1155.

31. **Burke J, Davison D & Hide W** (1999) *Genome Res.* **9**, 1135–1142.

32. **Miller G, Fuchs R & Lai E** (1997) *Genome Res.* **7**, 1027–1032.

33. **Lennon G, Auffray C, Polymeropoulos M & Soares MB** (1996) *Genomics*, **33**, 151–152.

34. **Ewing B & Green P** (1998) *Genome Res.* **8**, 186–194.

35. **Ewing B, Hillier L, Wendl MC & Green P** (1998) *Genome Res.* **8**, 175–185.

36. **Tatusov RL & Lipman DJ** (1998) *The DUST Program:* ftp://ftp.ncbi.nih.gov/pub/tatusov/dust/ (unpublished).

37. **Benson G** (1999) *Nucleic Acids Res.* **27**, 573–580.

38. **Green P** (1997) *CROSS_MATCH and PHRAP*. http://www.phrap.org/phredphrapconsed.html (unpublished).

★ 39. **DurhamAM, Kashiwabara AY, Matsunaga FT, *et al.*** (2005) *Bioinformatics*, **21**, 2812–2813. – *The original publication describing EGENE, an automated pipeline-generation system. The reference also describes the preferred protocol used in this chapter for EST pre-processing and assembly.*

40. **Gordon D, Abajian C & Green P** (1998) *Genome Res.* **8**, 195–202.

41. **Altschul SF, Madden TL, Schaffer AA, *et al.*** (1997) *Nucleic Acids Res.* **25**, 3389–3402.

★★★ 42. **WolfsbergT & Landsman D** (2001) In *Bioinformatics – A Practical Approach in the Analysis of Genes and Proteins.* Edited by A Baxevanis & B Ouellette. John Wiley & Sons, New York. pp. 283–301. – *A concise but very clear text describing the most important bioinformatic aspects of EST processing. Recommended for beginners.*

★★★ 43. **Hide W, Miller R, Ptitsyn A, Kelso J, Gopalakrishnan C & Christoffels A** (1999) In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology (ISMB99).* AAI Press, Menlo Park, CA. Electronic version vailable at http://bioinf.mpi-sb.mpg.de/conferences/ismb99/WWW/TUTORIALS/tutorial_6.html. – *An excellent text discussing the most important aspects of EST clustering.*

44. **Lee Y, Tsai J, Sunkara S, *et al.*** (2005) *Nucleic Acids Res.* **33**, D71–D74.

45. **Pertea G, Huang X, Liang F, *et al.*** (2003) *Bioinformatics*, **19**, 651–652.

46. **Pontius JU, Wagner L & Schuler GD (**2003) In *The NCBI Handbook.* Edited by J McEntyre & J Ostell. National Center for Biotechnology Information, Bethesda, MD, pp. 1–12 (Section 21).

47. **Ptitsyn A & Hide W** (2005) *BMC Bioinformatics*, **6** (Suppl. 2), S3.

48. **Liang F, Holt I, Pertea G, Karamycheva S, Salzberg SL & Quackenbush J** (2000) *Nucleic Acids Res.* **28**, 3657–3665.
49. **Marth GT, Korf I, Yandell MD, *et al.*** (1999) *Nat. Genet.* **23**, 452–456.
50. **Buetow KH, Edmonson MN & Cassidy AB** (1999) *Nat. Genet.* **21**, 323–325.
51. **Irizarry K, Kustanovich V, Li C, *et al.*** (2000) *Nat. Genet.* **26**, 233–236.
52. **Picoult-Newberg L, Ideker TE, Pohl MG, *et al.*** (1999) *Genome Res.* **9**, 167–174.
53. **Florea L, Hartzell G, Zhang Z, Rubin GM & Miller W** (1998) *Genome Res.* **8**, 967–974.
54. **Liu D & Graber JH** (2006) *BMC Bioinformatics*, **7**, 77.
55. **Pearson WR & Lipman DJ** (1988) *Proc. Natl. Acad. Sci. U.S.A.* **85**, 2444–2448.

★★ 56. **KorfI, Yandell M & Bedell J** (2003) *BLAST*. O'Reilly and Associates, Inc., Sebastopol, CA. *– A reference book on BLAST including numerous protocols for EST mapping to a genome, clustering, and annotation.*

57. **Mott R** (1997) *Comput. Appl. Biosci.* **13**, 477–478.
58. **Wheelan SJ, Church DM & Ostell JM** (2001) *Genome Res.* **11**, 1952–1957.
59. **Kent WJ** (2002) *Genome Res.* **12**, 656–664.
60. **Slater GS & Birney E** (2005) *BMC Bioinformatics*, **6**, 31.
61. **Hillier LD, Lennon G, Becker M, *et al.*** (1996) *Genome Res.* **6**, 807–828.
62. **Romani A, Guerra E, Trerotola M & Alberti S** (2003) *Nucleic Acids Res.* **31**, e17.
63. **Zhang C, Xie Y, Martignetti JA, Yeo TT, Massa SM & Longo FM** (2003) *DNA Cell Biol.* **22**, 303–315.
64. **Pruitt KD, Tatusova T & Maglott DR** (2005) *Nucleic Acids Res.* **33**, D501–D504.
65. **Sherry ST, Ward MH, Kholodov M, *et al.*** (2001) *Nucleic Acids Res.* **29**, 308–311.