

**Expressed sequence tags: analysis and annotation**

John Parkinson and Mark Blaxter

Institute of Cell, Animal and Population Biology

Ashworth Laboratories

King's Buildings

Edinburgh

EH9 3JT

UK

phone: +44 131 650 6760

fax: +44 131 650 7489

email: [mark.blaxter@ed.ac.uk](mailto:mark.blaxter@ed.ac.uk)

web: <http://www.nematodes.org>

## 1. Introduction: The special analysis problems associated with ESTs

Expressed sequence tags (ESTs) are single pass DNA sequence reads derived from cDNA clones (1, 2). The EST strategy has been used by many parasitology programmes for gene discovery, for drug target or vaccine candidate identification, with significant success (3-22). In addition, EST sequences, and the clones from which they derive are often the input reagents for other methodologies such as immunoscreening or DNA microarray expression analysis. The sheer size of many EST datasets makes human curation of the data difficult or impossible, and thus computer-based bioinformatic methods have been developed to reduce the complexity of the datasets and efficiently analyse them for informative content. The reduction in complexity in the number of different sequence objects being analysed is matched by an increase in usability of the data, where each gene is represented by a single sequence object with much higher quality sequence and annotation, and is essential for many downstream applications.

An EST dataset is a sample of the mRNAs present in the original tissue used for construction of the cDNA bank. As different genes have very different patterns of steady-state mRNA levels, an EST dataset will have some genes represented by many, many ESTs and others not represented at all. This differential representation is one of the benefits of EST analysis as it permits inference of the expression levels of the genes, as well as one of its major analytical problems. Placing the ESTs into clusters that are inferred to derive from one gene can be achieved by grouping those with high levels of identity, but even this step can be difficult. As ESTs are single pass reads, the actual sequence derived from the sequencing chromatograph may contain errors. In particular, the beginning and end of the sequence may be more prone to error (i.e. of lower quality) than the central portion. Different ESTs may therefore disagree in their sequence because of sequencing error, and error rates may differ along each sequence.

Most cDNA banks are constructed using technology that does not guarantee that every clone will be full length and different ESTs that derive from the same mRNA sequence may therefore have different 5' and 3' ends. The process of library construction can sometimes also result in the construction of chimaeric cDNAs that arise from the illegitimate ligation of two fragments deriving from different genes. Comparing the chimaeric EST with a nonchimaeric one, the sequences will appear to be identical for some of the sequence and then diverge significantly after the illegitimate join.

Unfortunately this phenotype is also a property of some real sequences. Many genes in eukaryotes give rise to alternatively spliced mRNAs where particular exons can be included or not depending on complex regulatory cues. An EST derived from an alternatively spliced version of a mRNA will also diverge significantly after the “join”.

When it comes to trying to attribute biological function to the genes that the ESTs represent, cross-species sequence comparisons are used. Prediction of open reading frames and encoded peptides from ESTs is compromised by the low quality of some of the sequence and in particular by the presence of insertions and deletions of bases that cause “virtual” frameshifts. Thus functional annotation of ESTs must recognise the possibility of poor sequence and frameshifts. In addition, as many parasitic organisms are only distantly related to well-studied model organisms, the evolutionary distance separating the parasite gene from its nearest sequence neighbour may be so great as to obscure informative similarity. Careful attention needs to be paid to sequence similarity searches using ESTs so that interpretation of a match is tempered by understanding of the evolutionary history of the organisms being compared.

EST sequencing is over ten years old (1, 2) and thus these problems are not new. A large number of software solutions have been developed for each one. What we present here is an integrated suite of software solutions to EST analysis that can be run on local computers. The reliance on open source or freely available software solutions, with easily customised parameters, allows us to propose a pipeline that can take ESTs from sequencer chromatograph to annotated database entry efficiently and cheaply (**Note 1**).

## 2. Materials

### 2.1 Computing hardware

The computing demands for analysis of even moderate-sized EST projects (~20,000 sequences) are not beyond the capabilities of today's modern desktop computers. The software we recommend is written for UNIX-based operating systems, and thus the bioinformatics machine should be capable of running the freely available LINUX version of UNIX. For the analyses outlined within this chapter we would recommend the following minimum setup:

PC with Pentium III or IV processor (800 MHz to 2.7 GHz),  
with 80GB of hard disk storage capacity and 512MB of RAM  
and an ethernet connection

By the time this book is published, such specifications will be superceded by entry level desktop personal computers costing a few hundred pounds (**Note 2**).

For an operating system, we recommend using Red Hat LINUX (current release 8.0) <http://www.redhat.com> although other flavours of LINUX should work equally well.

### 2.2 Software sources (**Note 3**)

LINUX, as with other UNIX systems, works most efficiently for you if you take close note of where files are kept, particularly "executable" files or programs. We suggest that core resources such as the **BLAST** and **phred** programs (see below) are stored in directories under `"/usr"` in the LINUX file heirarchy, such as `"/usr/local/bin"`, or `"/usr/ncbi"` (for BLAST). **perl** programs written by you or downloaded (for example from our world wide web site <http://www.nematodes.org/scripts/>) can be kept in a binary file directory `"/usr/local/bin/"`. The LINUX notation for the home directory of the current user is `"~"` (tilde). If you are organised about where programs are stored it is possible to set your operating system login to "know" where to look for them (**Note 4**).

2.2.1 Many of the programs mentioned in this chapter rely on the use of the **perl** scripting language (current version 5.8, although the scripts also work on the previous release 5.6), which is usually bundled with the LINUX operating system. **perl** is freely downloadable

from <http://www.perl.org/>.

2.2.2 **perl** scripts and programs written by us (**format\_traces.pl**, **trace2dbest**, **make\_a\_table.pl**, **fsa2clus**, **CLOBB.pl**, **pre\_assemble.pl**, **multi\_phrap**, **prepare4blast.pl**, **blast\_5db.pl**) are available at our web site <http://www.nematodes.org/scripts>. We recommend that you store these in “~/usr/local/bin/”.

2.2.3 The **trace2dbest** software package (current version 1.0-2; **Note 5**) is available as an rpm for ease of installation. An rpm is a software bundle that is easily installed onto a host machine using the RPM package manager that comes with Red Hat LINUX distributions, but the scripts can be made available separately by emailing the authors. To install the package on a machine running Red Hat LINUX simply download the file from the above site using the ftp capabilities of a world wide web browser. Once the file is downloaded, enter

```
rpm --install trace2dbest.pl-1.0-2.i386.rpm
```

The executables are located in “/usr/local/bin” (**Note 4**).

2.2.4 Our in-house EST clustering solution, **CLOBB** (23), is a freely available perl program that relies only on the installation of the “blastall” executable (obtained as part of the BLAST package; see below). It is available from <http://www.nematodes.org/scripts>.

2.2.5 **phred**, **phrap** and **cross\_match** are available via a license (free to academics) from <http://www.phrap.org> (24, 25). Install these programs in “/usr/local/bin/” (**Note 4**).

2.2.6 **BLAST** is freely available from the National Center for Biotechnology Information (NCBI) from their website <http://www.ncbi.nlm.nih.gov/BLAST/> (26, 27). We recommend storing the BLAST distribution in a directory “/usr/ncbi/” and the executables in “/usr/ncbi/bin” (**Note 4**).

### 3. Methods

In the following we use as an example an EST project from the platyhelminth *Echinococcus granulosus*, and presume that all analyses are being performed in a directory “~/ESTproject/” in your home directory.

#### 3.1 Preparing ESTs for database submission and downstream analysis

##### 3.1.1 Extracting DNA sequence from sequencer chromatograms and trimming low quality sequence

Different automated sequencers have their own, proprietary associated software suites for processing the chromatographic information and predicting the sequence of bases for each sequencing read. As these softwares are closely tied to each sequencing platform, and can be costly, they are not ideal. However, software for the calling of bases from the chromatograms has been developed and refined in the public genome projects and is available for local installation, and can read the chromatograms produced by all the major sequencers (**Note 6**). This allows a single user to process sequences according to their own criteria.

The program **phred** was developed by Phil Green and colleagues and is the “industry standard” (**Note 7**). **phred** takes the raw chromatograms and uses a series of heuristics to both infer which base should be called at any particular position, and to ascribe that base a quality score. The score is computed by assessing the relative height, separation and shape of each fluorophore peak in relation to the signal from other fluorophores. This score is on a log scale, and a **phred** score of 20 (i.e. an error probability of one in  $2^{20}$ ) is usually taken to indicate a good base call. The **phred** scores can then be used to remove low quality bases from the predicted sequence.

As each EST is usually sequenced from a vector-directed primer, each sequencing trace will include a small piece of vector DNA sequence. At the vector-insert junction there may also be sequence corresponding to linkers or adapters used in the construction of the cDNA bank. In the case of a short insert, the derived sequence may extend through the poly(A) tail into the vector on the other side. These sequences should also be trimmed from the EST before further analysis. It is usual to trim the poly(A) tail and note its presence in the EST submission data as a text comment. cDNA bank construction can also inadvertently result in the cloning of contaminant DNA, particularly DNA from

*Escherichia coli* and its bacteriophages that gets into the reactions through contamination of recombinant enzymes. These sequences should also be removed. A free program, **cross\_match**, is used to find and trim away any vector, linker/adaptor and contaminant sequences. A pattern-recognition script written in perl can be used to remove and log poly(A) stretches at the 3' end of sequences.

### 3.1.2 Submitting ESTs to the public databases

The public EST database, dbEST, is a valuable resource for genomics and genetics, and currently contains over 10<sup>7</sup> sequences (mostly from humans and other mammals) (see <http://www.ncbi.nlm.nih.gov/dbEST/index.html> and [http://www.ncbi.nlm.nih.gov/dbEST/dbEST\\_summary.html](http://www.ncbi.nlm.nih.gov/dbEST/dbEST_summary.html)) (28). While not all parasite EST projects deposit their data in dbEST, we would encourage sequencers to do so, as the benefits of collective access to such data are enormous. In particular, the presence of parasite EST data in public databases alerts researchers working on model organisms to the existence of homologues of their favourite genes in medically important taxa, and has resulted in many fruitful collaborations. The public databases provide a mechanism for submitting sequences and thus being allocated database accession numbers but preventing immediate public release. This mechanism allows researchers to have their data ready for release on one date, or on publication of a descriptive paper.

dbEST has a simple but rigid submission format for EST data (see [http://www.ncbi.nlm.nih.gov/dbEST/how\\_to\\_submit.html](http://www.ncbi.nlm.nih.gov/dbEST/how_to_submit.html) and Table 1). The format involves three files describing the cDNA bank from which the ESTs were derived, a contact person and a publication. The EST sequence is submitted in a fourth file that quotes these three descriptors. As the format of all four file types is simply text (or flat file format) it is simple to use perl scripts to generate EST submission files. Many researchers screen the databases using text-based searches (looking for annotation matches to “protease” or “kinase” for example) and thus adding some preliminary annotation information as a text comment in the EST submission is very useful. This can be simply achieved by performing a BLAST sequence similarity search (see below) against a protein database and using the top-scoring match definition line to add a “similar to ...” comment.

### 3.1.3 Automating sequence calling, trimming and preparation of dbEST submission files

To simplify the process of preparing EST sequences for submission to dbEST, we have developed a freely available perl based software package (**trace2dbest**) that uses phred, cross\_match and BLAST to batch process raw EST sequence trace files. The package includes two main programs: **format\_traces.pl** is a simple script which can be used to quickly used to reformat the names of the sequence traces to follow the suggested naming conventions (**Note 8**) and **trace2dbest.pl** is a text-based menu driven software package which creates the dbEST submittable files from the raw sequence traces.

3.1.3.1 In the following we presume that you have transferred the sequencer output files to your computer, and have stored them in a directory “~/ESTproject/traces”.

```
cd ~/ESTproject
```

3.1.3.2 The chromatograph files need to have a uniform naming scheme in order for the subsequent processes to find and recognise them. The perl script **format\_traces** renames files into our favoured format (**Note 8**). **format\_traces.pl** is run by specifying a series of arguments: “-dir” specifies which directory should be searched; “-add” specifies which text should be added to the front of each file name (eg the EST library identifier; **Note 8**); “-txt” indicates text that should be removed from filenames where it is found; “-sub” indicates text that should be replaced for the text removed using “-txt” and “-format” tells the program to reformat the filenames assuming a 96-well plate layout. Thus, if the files moved to “~/ESTproject/traces” above were named “1.seq” to “96.seq”, and we wished them named “Eg\_ad1\_01A01” to “Eg\_ad1\_01H12” (where Eg stands for *Echinococcus granulosus*, ad1 stands for adult library version 1 and the microtitre plate number is 01; see **Note 8**) we would enter:

```
/usr/local/bin/format_traces.pl -dir traces -add Eg_ad1_01 -  
txt .seq -format
```

(All this is entered as one line with no returns.)

3.1.3.3 To run **trace2dbest** enter the command

```
/usr/local/bin/trace2dbest.pl
```

If you have used the program before, you may be asked to clean up any directories which were previously created. Further instructions on the use of this package are provided as one of the menu options.

3.1.3.4 You will be offered a menu with a list of options to select from. The first task is to select the library used to create the sequences. If you are sequencing from a library for the first time, you will need to enter the library details by selecting either option “2” or “3” from the menu (see Figure 1 for a description of the sort of information needed for each library).

3.1.3.5 After the library has been selected you will be prompted to enter the name and path to the directory where all the traces are located (in the example here, this would be “~/ESTproject/traces”). `trace2dbest.pl` will only process sequencer trace files with the name specified by the EST library identifier, which is formed from the species name tag, and the library tag (see **Note 8** for information on naming conventions used).

3.1.3.6 You will then be asked to specify the sensitivity of vector trimming. This is very much related to the vector and primer combination used and we recommend that you set the sensitivity to “Low” unless you find that vector sequence is still present in the final submission files.

3.1.3.7 The next step asks if you'd like to perform some preliminary annotation to include as a text comment in the EST submission file. Although there is an option to use the NCBI remote blast client (installed as part of the NCBI BLAST package), due to the time associated with this procedure we recommend that you perform a BLASTX search against a locally installed copy of the nr protein database (see 3.4.1.7 below).

3.1.3.8 After selecting all of these options, the program will then use **phred** to convert the raw trace files into usable sequence data and **cross\_match** to perform vector trimming. Low quality data and poly(A) tails are then discarded and the optional BLAST step may be undertaken. A series of comments relating to the progress of the process will appear on screen.

3.1.3.9 The dbEST submission files (see Table 1) are created in a directory “subfiles” with the suffix “.sub”. These files may be concatenated using the simple UNIX command “`cat`” into one file which may be emailed to the dbEST repository (`batch_sub@ncbi.nlm.nih.gov`). For example:

```
cat ~/ESTproject/subfiles/*.sub ~/ESTproject/submission_file
```

3.1.3.9 After the trace2dbest process, the “~/ESTproject” directory, which used to have but a single directory “traces” in it, will have a number of additional files and directories, visible by using the “ls” command:

The results of “ls” on “~/ESTproject/” after running format\_traces.pl and trace2dbest

What this is:

*fasta/*

a directory of phred output sequence files in FASTA format: for each input file in /traces there will be two new files “xyz.seq” and “xyz.seqsc”

*logfile/*

a directory of logfiles of the trace2dbest process

*phd\_dir/*

a directory of phred output process files: for each input file in /traces there will be a new file “xyz.phd”

*qual/*

a directory of phred output quality files: for each input file in /traces there will be a new file “xyz.qual”

*scf/*

a directory of chromatograph files produced by phred: for each input file in /traces there will be a new file “xyz.scf”

*sequences*

a directory of trace2dbest trimmed sequence files: for each input file in /traces there will be a new file with the same name

*subfiles/*

a directory of trace2dbest EST submission files: for each input file in /traces there will be a new file “xyz.sub”

*submission\_file*

the result of the “cat” command, ready to send to dbEST

*traces/*

the directory of original trace files

### 3.2 Clustering of ESTs into putative genes

As explained above, many ESTs may derive from the same gene. It is therefore advisable to group the sequences on the basis of sequence similarity into clusters which can be used to derive consensus sequences. A cluster is defined as a unique set of sequences which share common sequence similarity. A cluster containing only one sequence is termed a singleton. Clustering both reduces the level of redundancy and increases the overall quality of the derived sequence. During production of ESTs, it is often useful to monitor the redundancy of the dataset to ensure that a particular cDNA library is not being over-sampled. Initially the level of redundancy from a newly sequenced library should be low but will increase as new ESTs are generated. From our experience, 10000 sequences from a good quality library should yield 4000 to 5000 clusters.

#### 3.2.1 CLOBB

Although there are a number of EST clustering solutions available (see 3.2.2 below), in order to monitor levels of sequence redundancy we have developed a custom clustering solution **CLOBB** (Cluster on Basis of BLAST) (23). An important feature of CLOBB is that cluster identifiers are retained between builds, allowing incremental analysis of an ongoing EST project.

3.2.1.1 Open a terminal and navigate to the project folder containing the sequences (in the example being followed this would be “~/ESTproject/”). CLOBB expects to find the processed EST sequences in a directory “sequences”. CLOBB documentation is available using the command

```
perldoc /usr/local/bin/CLOBB.pl
```

3.2.1.2 The program is run by specifying a three letter cluster identifier to be used for creation of the clusters. We recommend that the taxon name is used, and we usually use “C” as the third letter to identify the resulting analysis as a cluster. For a species “*Echinococcus granulosus*” the command could be:

```
cd ~/ESTproject/  
  
/usr/local/bin/CLOBB.pl EGC
```

3.2.1.3 The main output of the program is a single, multi-sequence FASTA (**Note 9**) format file beginning with the cluster identifier followed by “EST.fsa” (in the example above the file would be called “EGCEST.fsa”). In this file, each sequence has had added to its “>”

header line a cluster identifier beginning with the three letter code and followed by a five digit number: thus EGC00001, EGC00002 etc.

3.2.1.4 To convert this file into a list of separate cluster files we provide a **perl** program, **fsa2clus**. To run **fsa2clus**, you must specify the three letter cluster identifier. Thus for the above example the command would be:

```
/usr/local/bin/fsa2clus EGC
```

This creates a directory "Clus" which contains individual cluster files in FASTA format containing the sequences associated with each cluster (designated ABC00001, ABC00002 etc) and a single FASTA file called "singletons.fasta" which contains all the sequences which did not group with any other sequence.

3.2.1.5 A number of other files are created during the **CLOBB** process including "merge" which contains a list of all the clusters which have been merged into a single cluster (as a result of a common sequence spanning between two clusters) and "supercluster" which contains a list of clusters which are significantly related to each other, representing either alternative splice variants or chimaeric clones. These files are available as aids to further annotation of the sequences and to record the history of changes between subsequent builds of the clusters. A "ls" of the project directory at this stage will reveal:

The results of "ls" on  
"~/ESTproject/" after  
running CLOBB and fsa2clus

What this is:

```
Clus/  
EGCEST.fsa  
EGCEST.fsa.nin,  
EGCFSA.fsa.nsq,  
EGCEST.fsa.nhr  
fasta/
```

the cluster files created by fsa2clus  
the major CLOBB output file  
BLAST database files produced by the CLOBB process

```
logfile/  
merge  
OUT/  
phd_dir/
```

a directory of phred output sequence files in FASTA format: for each input file in /traces there will be two new files "xyz.seq" and "xyz.seqsc"  
a directory of logfiles of the trace2dbest process  
a CLOBB record of clusters merged during the process  
a directory of CLOBB logfiles  
a directory of phred output process files: for each input file in /traces there will be a new file "xyz.phd"

```
qual/
```

a directory of phred output quality files: for each input file "xyz" in /traces there will be a new file "xyz.qual"

```
scf/
```

a directory of chromatograph files produced by phred: for each input file "xyz" in /traces there will be a new file "xyz.scf"

```
seqfiles_done
```

a directory of sequence files that have been processed by

<i>sequences</i>	CLOBB: for each input file “xyz” in /sequences there will be two files, “xyz” and “xyz.old”
<i>subfiles/</i>	a directory of trace2dbest trimmed sequence files: for each input file “xyz” in /traces there will be a new file with the same name
<i>submission_file</i>	a directory of trace2dbest EST submission files: for each input file “xyz” in /traces there will be a new file “xyz.sub”
<i>supercluster</i>	the result of the “cat” command, ready to send to dbEST
<i>traces/</i>	a CLOBB record of clusters with significant similarity to each other
	the directory of original trace files

### 3.2.2 Other options

Other clustering solutions are available including StackPACK (29), ICAtools (30, 31) and the clustering suite of tools available from TIGR (see <http://www.tigr.org/tdb/tgi/software/>). However of these, only StackPACK (available at <http://www.sanbi.ac.za/Dbases.html>) like CLOBB, offers the ability to maintain cluster identifiers between subsequent builds. This is an important criterion in being able to monitor library quality/redundancy.

### 3.3 Predicting the consensus sequence for each EST cluster

Once the sequences have been grouped into clusters, the next step is to obtain consensus sequences for each cluster which contains more than one sequence. A number of programs are currently available which were originally developed to derive contiguous stretches of sequence (termed contigs) from genome sequencing initiatives. These programs may be readily applied to assemble contig sequences from clusters of ESTs. As the stringency for contig building differs between CLOBB and these programs, multiple contigs may result from each cluster. These may indicate the presence of alternative splices or alleles which were not initially detected during the clustering step. This assembly process, thus represents an additional clustering step. The use of two assembly programs, CAP3 and phrap, is outlined here.

**CAP3**, developed by Xiaoqiu Huang, makes use of base quality values (if available) in constructing an alignment of sequence reads and generating a consensus sequence for each contig. It clips poor 5' and 3' regions of reads and uses only good regions of reads in assembly. **phrap** is a program originally developed by Phil Green to assemble DNA shotgun sequence data (24, 32). phrap uses quality data generated during the base-calling step to create consensus sequences. It has the advantage over other programs such as CAP3 (33) in that built contigs include sequence data from regions of the alignment spanned by only a single sequence (i.e. end overhangs). For both processes, if a quality file is supplied it must have the same name as the cluster file with the additional suffix ".qual". Quality files are obtained from the "qual" directory created by phred in the base-calling procedure (as implemented in the trace2dbest program, 3.1.3 above).

If no quality file is supplied, the programs assume a relatively low score for the quality of each base.

3.3.1 To generate phrap-ready quality files from CLOBB output, we have developed a perl script called **pre\_assemble.pl**. In this script you specify the directory where the original ".qual" and sequence files are located. The program scans through the cluster files, identifies the appropriate quality and sequence files and concatenates them to an appropriately named cluster quality file: all of these are stored in a new directory called "phrap". If the files are not found on the system, then the program creates a false entry in the cluster quality file which each base is given a default quality value which may be defined by the user. Since the original sequence files are being used, they must be trimmed for poly(A) tails and vector contamination.

```
cd ~/ESTproject/  
/usr/local/bin/pre_assemble.pl
```

3.3.2 In most cases, it is preferable to run the assembly process as a batch process. To this end we have created a perl script called **multi\_phrap**:

```
cd ~/ESTproject/phrap/  
/usr/local/bin/multi_phrap EGC
```

where EGC represents the three letter cluster identifier. The program uses `cross_match` to remove any remaining vector contamination from the cluster sequence files and removes the corresponding quality value entries in the cluster quality file. It then uses `phrap` to assemble the clusters depending upon the selection made by the user. If no consensus sequence can be derived, a consensus sequence is generated by selecting the longest sequence from the cluster sequence file.

3.3.3 Performing a “`ls`” command at this stage reveals within directory “`phrap`” for each input file (such as “`EGC00001`”) there will be the following:

<code>EGC00001</code>	the original CLOBB-derived file moved by the <code>pre_assemble.pl</code> script
<code>EGC00001.ace</code>	part of the <code>phrap</code> output
<code>EGC00001.qual</code>	the original phred-derived file moved by the <code>pre_assemble.pl</code> script
<code>EGC00001.contigs</code>	a FASTA file of the contigs derived from <code>EGC00001</code>
<code>EGC00001.contigs.qual</code>	quality files associated with each contig
<code>EGC00001.single</code>	any singletons not used for building the contig
<code>EGC00001.log</code>	a logfile
<code>EGC00001.list</code>	a list of sequences used

### 3.4 BLAST-based sequence similarity analysis

Functional annotation of sequences via bioinformatic tools is an important first step in the utilisation of EST datasets for understanding the biology of parasites. Because genes evolve, it is possible to examine sequences from different species for conservation of sequence, and then use this conservation to transfer functional annotation from one gene of known role to an otherwise uncharacterised EST cluster. The simplest way of doing this is to use sequence comparison tools that are attuned to the evolutionary changes known to occur in genes and their encoded proteins to search the EST consensus sequence against a database of annotated sequences. The best tool for this is BLAST, the basic local alignment search tool (26, 27). BLAST searches for exact matches to short subsequences of the query and then tries to extend these initial “hits” using a set of parameters that give scores for matches and mismatches. The significance of the final best hit (called a high-scoring sequence pair or hsp in BLAST) is calculated based on the probability of finding, by chance, a hit as good as the one found, given a query sequence of the same residue composition and length, and a database of the same size and complexity. The output of a BLAST search is a table of matches, with both the score (given by summing the match rewards and mismatch penalties) and the probability of that score for each sequence with a hsp. Low complexity sequence (such as peptide or nucleotide repeats, or homo-residue runs) can generate spuriously high-scoring matches and thus it is usual to mask these parts of a sequence before searching. Masking of the sequence allows the program to make the comparison based on the higher complexity regions, and thus return hsp of more likely biological significance. The masked regions are indicated by “X” in the output. Filtering for low complexity is on by default.

The BLAST family of programs can compare nucleotide with nucleotide (called BLASTN), nucleotide with protein (by translating the nucleotide sequence in all six frames; BLASTX), protein with protein (BLASTP) and protein “back translated” to nucleotide against nucleotide (TBLASTN). It is also possible to compare the six possible protein sequences from a nucleotide query with a six-frame translation of a nucleotide database (TBLASTX). BLASTN is optimised for finding very close matches, and is less suited for comparisons across wide evolutionary distances. The protein-protein comparisons (BLASTP, BLASTX and TBLASTX) use a matrix that gives a set of scores for each substitution, and these matrices can be tuned for looking at even very distant similarity. For EST analysis the most informative searches are BLASTX, asking if there are known proteins with similarity to the

potential translation of the EST, and BLASTN, asking if the gene or one very closely related, has been sequenced before.

Web-based and client-server processes are available for performing all versions of BLAST search (for examples, see <http://www.ncbi.nlm.nih.gov/blast/> and <http://www.ebi.ac.uk/blast2/>). While these services are comprehensive, based on the complete, up-to-date public databases, they can be slow, particularly if many comparisons are required. For most purposes it is faster and just as informative to use BLAST locally, using customised databases extracted from the public resources of GenBank, EMBL and DDBJ.

### 3.4.1 Local BLAST databases

BLAST on a local machine can use either a remote database (client-server BLAST) or a local database. The use of local databases allows users to tailor searches to their needs, and in particular to perform searches restricted by taxonomy. Thus a search strategy for an EST project might include (1) a BLASTN search against nucleotide sequences from the same species (or the same genus) (2) a TBLASTX search against nucleotide sequences from the same major taxonomic group (class or phylum), excluding the species or genus of interest and (3) a BLASTX search against a nonredundant protein database from all organisms. Building these local databases is relatively easy, using the ENTREZ system provided by the National Center for Biotechnology Information (NCBI).

3.4.1.1 Launch a world wide web browser such as Netscape or Explorer and go to the NCBI home page <http://www.ncbi.nlm.nih.gov/>.

3.4.1.2 In the ENTREZ Search bar near the top of the page, change the database (default is "Nucleotide") to "Taxonomy" using the pull-down menu, and enter the name of the taxon of interest (for example "Echinococcus granulosus"). Click on the "Go" button. The database server at NCBI will look for the term entered and return a page with matched entries: select the taxon you are interested in and navigate until you have the page listing its attributes as recorded by NCBI. This taxonomy database entry will start with a Taxonomy ID code, a number, which you should record (**Note 10**).

3.4.1.3 Select, from the small table on the right of the page the dataset you would like to download (nucleotide or protein) (**Note 11**). The ENTREZ system will return you a page showing the first 20 sequences of the set selected. In the top ENTREZ search bar you will see the taxonomy ID of your chosen taxon, with a qualifier "[Organism]". (There may be additional qualifiers after organism, but they are not relevant here.).

3.4.1.4 In the "Display" bar, change the format you wish to see from "Summary" to "FASTA" (**Note 9**), and the "Send to" option from "Text" to "File". Click on the "Send to" button.

3.4.1.5 The browser will ask if you want to download all the sequences selected. Reply yes. It will then display the first sequence in FASTA format, and a dialog will appear asking you where the program should save the file, and what name it should be given. We would recommend that you use a name that is informative of the taxon source, the sequence

type and the format (such as “E\_granulosus\_nuc.fsa”). The sequences will be transferred to your computer. In addition, it is best to save all BLAST database files in a single location so that you do not have to remember where each one is. We suggest making a “localdb” directory in your home directory (see below).

3.1.4.6 To get databases with a more complex inclusion of sequences, use the ENTREZ system at the NCBI. Go to the NCBI home page, and in the ENTREZ Search bar at the top enter the query you wish to search. You can combine searches with “OR” or “NOT”. Thus to generate a database of *Echinococcus* (txid 6209) and *Schistosoma* (txid 6181) sequences you could enter “txid6209[Organism] OR txid6181[Organism]”. To build a database of platyhelminth (txid 6157) sequences excluding *Echinococcus* the query would be “txid6157[Organism] NOT txid6209[Organism]”. Select the database from which you want to download (“Protein” or “Nucleotide”) and then follow the same procedure as given in 3.1.4.5 above.

3.4.1.6 The sequence file now has to be properly formatted for BLAST. A utility for this is provided with the BLAST programs. Launch a terminal window. Use the mkdir command to make a directory called “localdb” in your home directory and move the FASTA file(s) of sequences you have downloaded there. Type

```
cd ~/
mkdir localdb
mv *.fsa localdb/
```

(This presumes you downloaded the sequences into the top level of your home directory.)

```
cd ~/localdb
```

Now run the **formatdb** command from the NCBI BLAST distribution. You use the “-i” modifier to tell it which file to process and the “-p” modifier to tell it whether the database is protein (“-p T”) or nucleotide (“-p F”). For example, using the *Echinococcus* dataset downloaded above, one would type:

```
/usr/ncbi/bin/formatdb -i E_granulosus_nuc.fsa -p F
```

If you now list (the ls command) the contents of the directory, you will find that in addition to the original “.fsa” file there are now “.fsa.nin”, “.fsa.nsq” and “.fsa.nhr” for nucleotide databases, or “.fsa.pin”, “.fsa.psq” and “.fsa.phr” for protein databases. These additional

files are indices used by BLAST to find matches. Repeat the formatdb process for each database downloaded.

3.4.1.7 The nonredundant protein database provided by the NCBI is very useful. It is generated by including only one representative of each set of proteins that are identical in sequence, and is much smaller than the whole protein database. This nr protein database is computed by NCBI frequently and is available from their ftp site. In a web browser, go to <ftp://ftp.ncbi.nih.gov/blast/db>. The nonredundant protein database is stored as "nr.Z": select this and download (**Note 12**). Open a terminal, move "nr.Z" to the localdb directory and uncompress it using the gunzip command. Format the database as you would any other.

```
mv nr.Z ~/localdb/  
  
cd localdb  
  
gunzip nr.Z  
  
rm nr.Z  
  
mv nr nr.fsa  
  
/usr/ncbi/bin/formatdb -i nr.fsa -p T
```

### 3.4.2 Basic BLAST

The BLAST family of programs is easily run interactively from the command line, and can also be called by perl programs. The ability to call for BLAST to process a file identified by a perl program allows the user to perform thousands of custom BLAST searches robotically, a huge saving in time. It is useful to understand the possible options for command-line operation of the BLAST family of programs (**Note 13**).

3.4.2.1 Open a terminal. The basic command line interface for BLAST is

```
/usr/ncbi/bin/blastall -p program -d database -i query -o  
outfile
```

(All this is entered as one line with no returns.)

The “-p” argument allows you to choose from one of the five flavours of BLAST (blastn, blastp, blastx, tblastn or tblastx; note that the program names are entered in lower case). The “-d” argument asks you for the location of the database you want to search (such as “~/localdb/E\_granulosus\_nuc.fsa”). The “-i” argument identifies the sequence you wish to compare, known as the query sequence. The query should be a text file in FASTA format (**Note 9**). The “-o” option tells the program what filename to write the file of results to. Additional arguments are available, including ones that set the program running with parameters (such as substitution matrix) different from those set as default (**Note 14**). One useful one to know is “-T”. Using “-T T” will yield hypertext-marked up output (html), ready for viewing in a web browser, while “-T F” will produce plain text output.

Thus, a command to search a nucleotide sequence, EGC00001, against the *Echinococcus* database, searching for matches due to encoded proteins might read

```
/usr/ncbi/bin/blastall -p tblastx -d  
~/localdb/E_granulosus_nuc.fsa -i EGC00001 -o EGC00001.out  
-T F
```

(All this is entered as one line with no returns.)

### 3.4.3 Running multiple BLAST searches

To run BLAST searches of a large number of sequences against a database you can either catenate all the sequences together into one large FASTA file and use it as the

query, or use a perl script that will take each file and perform a search. In the first case, BLAST understands that a multiple-sequence FASTA file should be searched as a series of individual sequences, but saves the results of all the searches to one (possibly very big) file. In the second, BLAST will save one search for each file processed. We have written a perl script (**blast\_5db.pl**) that will take all the sequence files in a single directory and perform up to five BLAST searches (all the same BLAST type) against different databases, saving the results as either plain text or html (see <http://www.nematodes.org/scripts>).

To facilitate BLAST analysis of the output of the EST sequence processing and clustering process outlined above we have written perl scripts that prepare the cluster contigs for BLAST analysis (**prepare4blast.pl**) and perform the BLAST searches (**blast\_5db.pl**)

3.4.3.1 Open a terminal, and in the “~/ESTproject/” directory, run prepare4blast.pl. The script assumes that you have completed the trace2dbest, CLOBB and consensus sequence prediction procedures outlined above, and collects the required sequences from the project directory into a new directory “~/ESTproject/blast”.

```
cd ~/ESTproject
/usr/local/bin/prepare4blast.pl
```

3.4.3.2 Decide which databases you wish to search and which variety of BLAST you wish to use. The program “blast\_5db.pl” only does one sort of BLAST at a time, on up to five databases. Locate the databases and check that they have been formatted. blast\_5db.pl is instructed where to look for sequences and databases from the command line as follows: the first argument is which program to use, the second is where the sequences are to be found, the third is what format of output is required (“H” for html, “T” for text) and these are followed by a list of the databases to be searched. Thus to search a nucleotide database in “~/localdb/” using BLASTN from CLOBB cluster consensus sequences in directory “~/ESTproject/blast/sequences” (as collected by prepare4blast.pl), with html output the commands would be:

```
cd ~/ESTproject/blast/
/usr/local/bin/blast_5db.pl blastn sequences H
~/localdb/E_granulosus_nuc.fsa
```

(All the BLAST command is entered as one line with no returns. In this example, only one database is searched: it is not necessary to give the program five databases every time. To search additional databases, enter the path to each database at the end of the command.)

3.4.2.3 The program will go through the sequences in the specified directory and use BLAST to search each against the specified databases, saving the results in directories named "databasename\_searchprogram", named after the sequence but with the extension ".html" for html files and ".txt" for text files.

### 3.4.3 Parsing BLAST outputs

As indicated above, the output from a BLAST search is a table of hsp matches for the query, ranked by the quality of the match. As with the actual carrying out of the BLAST searches, reviewing the results can be tedious if carried out piecemeal. The standard format of the BLAST output allows us to use text-processing algorithms to extract the relevant information and present it in more palatable form. Below we describe the use of a simple method for extracting significant matches from a set of BLAST results and tabulating them in a web-browser readable (html) format. This sort of simple extraction of data is most useful for medium-size datasets (up to ~1000 ESTs or clustered EST consensus sequences). Because this method only looks at the top hit for each search, informative or intriguing matches scoring less highly will not be noted. For a fuller analysis of the BLAST results a relational database is most useful, as described in 3.6 below.

A BLAST search output has the following features (see Table 2): (1) a header indicating which program was used (2) identifiers for the query and the database, (3) a table of hsps and (4) alignments of the query with the hsps. These features can be changed by using options in the BLAST command, but in the following we assume that the default settings are used. The table of matches and the alignments contain most useful information. In the table each match is listed with a segment of the FASTA ">" definition line as descriptor, the raw (bit) score and the E (expect) value. The table is usually sorted by E value, though it can be reported sorted by score. There are (unfortunately) no hard and fast rules for what constitutes a real, or biologically significant hsp match. Generally, in protein-protein comparisons (i.e. BLASTP, BLASTX and TBLASTX) a score of >80 bits and an E value of less than  $e^{-6}$  is considered necessary before a hsp would be examined further. Matches scored at between  $e^{-6}$  and  $e^{-8}$  are often to short domains, but are not difficult to come upon by chance. In nucleotide-nucleotide comparisons, one is searching for bit scores of >400 and E values of less than  $e^{-50}$ .

The table of significant hsps always starts with the text "Sequences producing significant alignments", and this can be used as a tag to identify where a process should start looking for the top hit. We have written a perl program (**make\_a\_table.pl**) that builds html-marked up tables of BLAST searches (see Figure 3 for an example). The perl program, on being told where to find the relevant sequence and BLAST result files, extracts from each the useful information and formats it for the world wide web. Each cell is hyperlinked to the

original BLAST result file (or sequence) and it is simple to screen down the table examining significant matches and ignoring the searches that returned no significant hits.

3.4.3.1 Open a terminal and navigate to the directory where your BLAST searches were carried out (“blast” from the example above). List the contents of the directory using the “ls” command to remind you of the names of the BLAST output directories.

3.4.3.2 Run the perl program make\_a\_table.pl. The program needs to know what the project name is (to place in the header and title of each html table), where the sequence files are and where each blast search output is to be found. It is set by default to split the results into groups of fifty sequences, as some web browsers have problems displaying html tables longer than about 50 rows.

Thus, using the mock example above (“~/bin/blast\_5db.pl blastn sequences H ~/localdb/E\_granulosus\_nuc.fsa”; The results of the “ls” command are shown in italics below.)

```
cd ~/ESTproject/blast/  
ls  
  
    proteins_1_blastx  
    proteins_2_blastx  
    sequences  
  
~/bin/make_a_table.pl Project_name sequences  
  
                                E_granulosus_nuc_blastn
```

(All this is entered as one line with no returns. You can enter up to five directories containing BLAST output at the end of the command. The program will return some processing comments.)

```
ls  
  
    Project_name_top_page.html  
    Project_name_table1.html  
    Project_name_table2.html  
    proteins_1_blastx
```

*proteins\_2\_blastx*

*sequences*

3.4.3.3 Open the document “Project\_name\_top\_page.html” in a web browser. It will have a title and links to the table pages, which will look like Figure 3.

### 3.5 Advanced analyses

#### 3.5.1 Relational databases for storing EST datasets

For large scale EST projects, it may not be appropriate to scan through many tables of html output looking for specific genes of interest. Furthermore, you may wish to undertake large scale analyses to identify groups of sequences sharing specific criteria (e.g. similar expression profiles). To help facilitate the use of larger datasets, we recommend the use of a relational database. Relational databases organise data into sets of tables related to each other through common values. Queries can then be readily formulated to extract data of interest from these tables. The more popular database schemes tend to be based on the structured query language (SQL). Many different public domain and commercial SQL database solutions exist including MySQL (<http://www.mysql.com/>) and Oracle (<http://www.oracle.com/>). Other non-SQL database solutions can also be used to store your data (e.g. Filemaker: <http://www.filemaker.com/>, Microsoft Access: <http://www.microsoft.com/office/access/default.asp> and ACeDB: <http://www.acedb.org/>). However, due to its performance and cost (public domain), we recommend the use of PostgreSQL (latest version 7.3.2). The PostgreSQL website (<http://www.postgresql.org/>) contains full instructions on how to download and implement PostgreSQL on your own workstation.

Once you have successfully installed PostgreSQL on your system, you will need to create a database to store your data. It is beyond the scope of this chapter to describe the full implementation of an EST project database. As a minimum, however, we recommend the use of three tables: a "cluster" table to store information on the consensus sequences and number of ESTs associated with each cluster; an "EST" table containing information on each EST (origin, sequence etc.); and a "BLAST" table containing information extracted from the BLAST searches associated with each cluster. perl scripts can then be written using the 'Pg' module (installed as standard with perl) to automatically populate the database from the previously generated flat files.

If you wish to serve the data to the wider community, then you may wish to set up a web server. Of the many flavours of web server available, Apache (current version 2.0) (<http://www.apache.org/>), is free and offers a well documented and supported solution. html pages are relatively easy to write and by incorporating cgi-scripts or using the embedded web scripting language php (<http://www.php.net/>) remote users can connect

and query your database (for an example see NEMBASE:  
<http://www.nematodes.org/nematodeESTs/nembase.html>).

Given the wealth of sequence data that EST projects generate, it is often useful to have tools which are able to identify sequences with unique properties. Relational databases aid such analyses and we currently implement a number of strategies to identify genes of interest. The simplest of these involves extracting the text from the definition lines of the BLAST output obtained for the clusters and using the database to search for clusters which display specific keywords (e.g. "kinase" or "cysteine protease"). Another approach involves the use of sequence similarity to identify particular gene families. This involves generating a BLAST-formatted database from the clusters and searching against a sequence of interest. Expression profiles can also be used as a method for obtaining genes of interest. If your EST dataset is derived from several libraries, it is possible to derive those clusters which have a particular pattern of expression, such as "Which clusters contain less than X ESTs from library A and more than Y ESTs from library B?".

### **3.5.2 SimiTri - a viewer for analysis of similarity datasets**

A novel approach to identifying interesting sequences from the dataset involves the use of similarity profiles. **SimiTri** (available at <http://www.nematodes.org/SimiTri>) is a java based tool which was developed to display the phylogenetic profiles for a large number of clusters on one graphic (34). For each cluster, the consensus sequence is BLAST searched against three different databases. BLAST bit scores in excess of 50 (a relaxed "significance cutoff") are extracted and used to compute the relative position of the cluster within triangular phase space. SimiTri provides a graphical view of this data, enabling the identification of sequences which possess similarity profiles which deviate from the norm (suggesting an atypical mode of evolution). SimiTri is freely available from <http://www.nematodes.org/SimiTri>, information on its use and implementation is found in the file SimiTri.txt.

### **3.5.3 Predicting proteins from EST consensus sequences**

Since ESTs represent the expressed portion of a genome, many of them will encode for proteins. You may therefore wish to identify these proteins to enable further downstream analyses. The prediction of proteins from EST derived sequence data is not a trivial exercise. The main problem that arises comes from the fact that ESTs are unverified sequences and are therefore prone to base-calling and sequencing errors. This can lead to frameshift errors when attempting to predict the peptide sequence. At present at three

software packages, ESTscan (35), Decoder (36), and DIANA-EST (37) have been developed which attempt to tackle this issue. The following are our current rules for best practice:

1. If there is close homologue in the protein database, then use the prediction from the BLASTX hit as an initial starting point for the prediction.
2. If there is no homologue, then you will need to use one or more of the programs outlined above, all of which have their advantages and disadvantages. We recommend the use of DECODER, which involves the use of the phred-derived quality scores.
3. In certain cases where the predictions do not cover large portions of the sequence under consideration, you may wish to undertake a simple 6-frame translation to 'fill-in' the missing areas.

**Table 1: Submission files for ESTs \***

<b>PUBLICATION</b>	
TYPE:	Pub **
MEDUID:	Medline unique identifier
TITLE:	Title of article **
AUTHORS:	Author name **
JOURNAL:	Journal name
VOLUME:	Volume number
SUPPL:	Supplement number
ISSUE:	Issue number
I_SUPPL:	Issue supplement number
PAGES:	Pages
YEAR:	Year of publication **
STATUS:	1=unpublished, 2=submitted, 3=in press, 4=published **
<b>LIBRARY</b>	
TYPE:	Lib **
NAME:	Name of library **
ORGANISM:	Scientific name of organism **
STRAIN:	Organism strain
CULTIVAR:	Plant cultivar
SEX:	Sex of organism (female, male, hermaphrodite)
ORGAN:	Organ name
TISSUE:	Tissue type
CELL_TYPE:	Cell type
CELL_LINE:	Name of cell line
STAGE:	Developmental stage
HOST:	Laboratory host
VECTOR:	Name of vector.
V_TYPE:	Type of vector (Cosmid, Phage, Plasmid, YAC, other)
RE_1:	Restriction enzyme at site1 of vector
RE_2:	Restriction enzyme at site2 of vector
DESCR:	Free text description of library preparation methods, vector, etc. Text starts on the line below the DESCR: tag.
<b>CONTACT</b>	
TYPE:	Cont **
NAME:	Name of contact person submitting the EST **
FAX:	Fax number as string of digits.
TEL:	Telephone number as string of digits.
EMAIL:	E-mail address
LAB:	Laboratory providing EST.
INST:	Institution name
ADDR:	Address string, comma delineation.

**Table 1 (continued)**

<b>SEQUENCE</b>	
TYPE:	EST **
STATUS:	"New" or "Update" **
CONT_NAME:	Name of contact ** §
CITATION:	Publication information ** §§
LIBRARY:	Library name ** §§§
EST#:	EST identifier assigned by contact lab **
CLONE:	Clone identifier
SOURCE:	Institutional source of clone e.g. ATCC
OTHER_EST:	Other ESTs from this clone
PCR_F:	Forward PCR primer sequence
PCR_B:	Backward PCR primer sequence
INSERT:	Insert length (in bases)
ERROR:	Estimated error in insert length (bases)
PLATE:	Plate number or code
ROW:	Row number or letter
COLUMN:	Column number or letter
SEQ_PRIMER:	Sequencing primer description or sequence
P_END:	Which end sequenced e.g. 5'
HIQUAL_START:	First base of highest quality sequence
HIQUAL_STOP:	Last base of highest quality sequence
DNA_TYPE:	cDNA **
PUBLIC:	Date of public release **
PUT_ID:	Putative identification of sequence by submitter
POLYA:	Y or N
COMMENT:	Comments about EST. Starts on line below COMMENT: tag
SEQUENCE:	Sequence starts on line below SEQUENCE: tag **

\* Based on the official dbEST submission instructions at

[http://www.ncbi.nlm.nih.gov/dbEST/how\\_to\\_submit.html](http://www.ncbi.nlm.nih.gov/dbEST/how_to_submit.html). The full version of the submission formats includes additional fields unlikely to be relevant for "neglected" parasitic organisms.

\*\* These tags are obligatory. All other tags are optional, but we would recommend that as many are filled as is possible.

§ The CONT\_NAME must match the NAME in the Cont submission.

§§ The CITATION must match the TITLE in the Pub submission.

.§§§ The LIBRARY must match the NAME in the Lib submission

**Table 2: BLASTX search output\***

BLASTX output information	Comment																																																																		
<p><b>BLASTX 2.0.11 [Jan-20-2000]</b></p> <p><b>Reference:</b>                      Altschul, Stephen F., Thomas L. Madden, Alejandro A. Schäffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman (1997), "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs", <i>Nucleic Acids Res.</i> 25:3389-3402.</p>	<p>a description of the program used and the literature reference for the program</p>																																																																		
<p><b>Query=</b> HCC00002.Contig1                      (1446 letters)</p> <p><b>Database:</b> swall-1; swall-2                      1,018,951 sequences; 323,619,622 total letters</p> <p>Searching.....done</p>	<p>a description of the query sequence and the database searched</p>																																																																		
<p>Sequences producing significant alignments:</p> <table border="0"> <thead> <tr> <th></th> <th style="text-align: right;">Score</th> <th style="text-align: right;">E</th> </tr> <tr> <th></th> <th style="text-align: right;">(bits)</th> <th style="text-align: right;">Value</th> </tr> </thead> <tbody> <tr><td>Q25049 Q25049 CAA56353.1 CAA56353.1 CAA56353.1 CAA56353.1 ...</td><td style="text-align: right;">880</td><td style="text-align: right;">0.0</td></tr> <tr><td>Q9TX76 Q9TX76 Desc: Beta-tubulin.</td><td style="text-align: right;">878</td><td style="text-align: right;">0.0</td></tr> <tr><td>Q25024 Q25024 AAA29170.1 Desc: Beta-tubulin.</td><td style="text-align: right;">877</td><td style="text-align: right;">0.0</td></tr> <tr><td>Q9GT34 Q9GT34 AAM95343.1 AAM95346.1 AAM95340.1 AAM95341.1 ...</td><td style="text-align: right;">873</td><td style="text-align: right;">0.0</td></tr> <tr><td>Q26901 Q26901 CAA93249.1 Desc: Beta-tubulin.</td><td style="text-align: right;">872</td><td style="text-align: right;">0.0</td></tr> <tr><td>Q9GT35 Q9GT35 AAG13954.1 AAK72123.1 AAM95347.1 Desc: Beta-...</td><td style="text-align: right;">872</td><td style="text-align: right;">0.0</td></tr> <tr><td>Q8MV65 Q8MV65 AAM95344.1 Desc: Beta-tubulin Cyca-1b.</td><td style="text-align: right;">872</td><td style="text-align: right;">0.0</td></tr> <tr><td>Q8MV63 Q8MV63 AAM95349.1 Desc: Beta-tubulin Cci-1b.</td><td style="text-align: right;">871</td><td style="text-align: right;">0.0</td></tr> <tr><td>Q9GT32 Q9GT32 AAG13961.1 Desc: Beta-tubulin isoform 1-3.</td><td style="text-align: right;">871</td><td style="text-align: right;">0.0</td></tr> <tr><td>Q9N611 Q9N611 AAF26294.1 AAF26293.1 Desc: Beta-tubulin.</td><td style="text-align: right;">870</td><td style="text-align: right;">0.0</td></tr> <tr><td>Q8MV60 Q8MV60 AAM95353.1 Desc: Beta-tubulin Ccr-1b.</td><td style="text-align: right;">870</td><td style="text-align: right;">0.0</td></tr> <tr><td>Q8MV62 Q8MV62 AAM95350.1 Desc: Beta-tubulin Cce-1a.</td><td style="text-align: right;">870</td><td style="text-align: right;">0.0</td></tr> <tr><td>Q8MV64 Q8MV64 AAM95345.1 Desc: Beta-tubulin Cyca-2a.</td><td style="text-align: right;">869</td><td style="text-align: right;">0.0</td></tr> <tr><td>Q8MV61 Q8MV61 AAM95351.1 Desc: Beta-tubulin Cce-1b.</td><td style="text-align: right;">869</td><td style="text-align: right;">0.0</td></tr> <tr><td>Q8MV66 Q8MV66 AAM95338.1 Desc: Beta-tubulin Cyp-1a.</td><td style="text-align: right;">867</td><td style="text-align: right;">0.0</td></tr> <tr><td>Q9GT33 Q9GT33 AAG13960.1 Desc: Beta-tubulin isoform 1-2.</td><td style="text-align: right;">862</td><td style="text-align: right;">0.0</td></tr> <tr><td>Q26900 Q26900 AAA30100.1 Desc: Beta-tubulin.</td><td style="text-align: right;">838</td><td style="text-align: right;">0.0</td></tr> <tr><td>Q25022 Q25022 AAA29168.1 Desc: Beta-tubulin.</td><td style="text-align: right;">836</td><td style="text-align: right;">0.0</td></tr> <tr><td>Q25023 Q25023 AAA29169.1 Desc: Beta-tubulin.</td><td style="text-align: right;">835</td><td style="text-align: right;">0.0</td></tr> <tr><td>Q18817 Q18817 CAB00853.1 Desc: C54C6.2 protein.</td><td style="text-align: right;">830</td><td style="text-align: right;">0.0</td></tr> </tbody> </table>		Score	E		(bits)	Value	Q25049 Q25049 CAA56353.1 CAA56353.1 CAA56353.1 CAA56353.1 ...	880	0.0	Q9TX76 Q9TX76 Desc: Beta-tubulin.	878	0.0	Q25024 Q25024 AAA29170.1 Desc: Beta-tubulin.	877	0.0	Q9GT34 Q9GT34 AAM95343.1 AAM95346.1 AAM95340.1 AAM95341.1 ...	873	0.0	Q26901 Q26901 CAA93249.1 Desc: Beta-tubulin.	872	0.0	Q9GT35 Q9GT35 AAG13954.1 AAK72123.1 AAM95347.1 Desc: Beta-...	872	0.0	Q8MV65 Q8MV65 AAM95344.1 Desc: Beta-tubulin Cyca-1b.	872	0.0	Q8MV63 Q8MV63 AAM95349.1 Desc: Beta-tubulin Cci-1b.	871	0.0	Q9GT32 Q9GT32 AAG13961.1 Desc: Beta-tubulin isoform 1-3.	871	0.0	Q9N611 Q9N611 AAF26294.1 AAF26293.1 Desc: Beta-tubulin.	870	0.0	Q8MV60 Q8MV60 AAM95353.1 Desc: Beta-tubulin Ccr-1b.	870	0.0	Q8MV62 Q8MV62 AAM95350.1 Desc: Beta-tubulin Cce-1a.	870	0.0	Q8MV64 Q8MV64 AAM95345.1 Desc: Beta-tubulin Cyca-2a.	869	0.0	Q8MV61 Q8MV61 AAM95351.1 Desc: Beta-tubulin Cce-1b.	869	0.0	Q8MV66 Q8MV66 AAM95338.1 Desc: Beta-tubulin Cyp-1a.	867	0.0	Q9GT33 Q9GT33 AAG13960.1 Desc: Beta-tubulin isoform 1-2.	862	0.0	Q26900 Q26900 AAA30100.1 Desc: Beta-tubulin.	838	0.0	Q25022 Q25022 AAA29168.1 Desc: Beta-tubulin.	836	0.0	Q25023 Q25023 AAA29169.1 Desc: Beta-tubulin.	835	0.0	Q18817 Q18817 CAB00853.1 Desc: C54C6.2 protein.	830	0.0	<p>a table of the high scoring sequence pairs (hsps) giving a brief identification (from the FASTA "&gt;" header), the score of each match and the computed probability (E value)</p>
	Score	E																																																																	
	(bits)	Value																																																																	
Q25049 Q25049 CAA56353.1 CAA56353.1 CAA56353.1 CAA56353.1 ...	880	0.0																																																																	
Q9TX76 Q9TX76 Desc: Beta-tubulin.	878	0.0																																																																	
Q25024 Q25024 AAA29170.1 Desc: Beta-tubulin.	877	0.0																																																																	
Q9GT34 Q9GT34 AAM95343.1 AAM95346.1 AAM95340.1 AAM95341.1 ...	873	0.0																																																																	
Q26901 Q26901 CAA93249.1 Desc: Beta-tubulin.	872	0.0																																																																	
Q9GT35 Q9GT35 AAG13954.1 AAK72123.1 AAM95347.1 Desc: Beta-...	872	0.0																																																																	
Q8MV65 Q8MV65 AAM95344.1 Desc: Beta-tubulin Cyca-1b.	872	0.0																																																																	
Q8MV63 Q8MV63 AAM95349.1 Desc: Beta-tubulin Cci-1b.	871	0.0																																																																	
Q9GT32 Q9GT32 AAG13961.1 Desc: Beta-tubulin isoform 1-3.	871	0.0																																																																	
Q9N611 Q9N611 AAF26294.1 AAF26293.1 Desc: Beta-tubulin.	870	0.0																																																																	
Q8MV60 Q8MV60 AAM95353.1 Desc: Beta-tubulin Ccr-1b.	870	0.0																																																																	
Q8MV62 Q8MV62 AAM95350.1 Desc: Beta-tubulin Cce-1a.	870	0.0																																																																	
Q8MV64 Q8MV64 AAM95345.1 Desc: Beta-tubulin Cyca-2a.	869	0.0																																																																	
Q8MV61 Q8MV61 AAM95351.1 Desc: Beta-tubulin Cce-1b.	869	0.0																																																																	
Q8MV66 Q8MV66 AAM95338.1 Desc: Beta-tubulin Cyp-1a.	867	0.0																																																																	
Q9GT33 Q9GT33 AAG13960.1 Desc: Beta-tubulin isoform 1-2.	862	0.0																																																																	
Q26900 Q26900 AAA30100.1 Desc: Beta-tubulin.	838	0.0																																																																	
Q25022 Q25022 AAA29168.1 Desc: Beta-tubulin.	836	0.0																																																																	
Q25023 Q25023 AAA29169.1 Desc: Beta-tubulin.	835	0.0																																																																	
Q18817 Q18817 CAB00853.1 Desc: C54C6.2 protein.	830	0.0																																																																	
<p>Q25049 Q25049 CAA56353.1 CAA56353.1 CAA56353.1 CAA56353.1 Desc: TUB-1 gene exon 1.                      Length = 448</p> <p>Score = 880 bits (2248), Expect = 0.0                      Identities = 427/448 (95%), Positives = 428/448 (95%)                      Frame = +2</p> <p>Query: 41 MREIVHVQAGQCQNQIGSKFWEVISDEHGIQPDGTYKGESDLQLERINVYYNEAHGGKYV 220                      MREIVHVQAGQCQNQIGSKFWEVISDEHGIQPDGTYKGESDLQLERINVYYNEAHGGKYV                      Sbjct: 1 MREIVHVQAGQCQNQIGSKFWEVISDEHGIQPDGTYKGESDLQLERINVYYNEAHGGKYV 60</p> <p>Query: 221 PRAVLVDLEPGTMDSVRSRGPYQGLFRPDNYVFGQSGAGNNWAKGHYTEGAELVDNVLDDV 400                      PRAVLVDLEPGTMDSVRSRGPYQGLFRPDNYVFGQSGAGNNWAKGHYTEGAELVDNVLDDV                      Sbjct: 61 PRAVLVDLEPGTMDSVRSRGPYQGLFRPDNYVFGQSGAGNNWAKGHYTEGAELVDNVLDDV 120</p> <p>(etc etc...)</p>	<p>for each hsp an alignment is given, with scores and percent identities (and similarities for protein searches)</p>																																																																		

\* Other varieties of BLAST yield very similar output

## Figure Legends

### Figure 1: Overview of the EST analysis process

Taking primary sequence reads through to an analysed dataset requires but a few steps that can be automated to handle thousands of sequences at a time. See the text for details of each step.

### Figure 2: Clustering using CLOBB

The process used by CLOBB to cluster ESTs relies on the use of custom-built BLAST databases. Each sequence is taken in turn and compared to all those previously examined. If a significant match is found, the cluster identifier of the matched sequence is attributed to the new one. If the new sequence is apparently novel, it is ascribed the next available cluster identifier. The CLOBB software is able to deal with more complex issues that can arise, such as a single new sequence matching more than one previous cluster (implying that clusters should be joined).

### Figure 3: An example of a table of EST blast results generated with `make_a_table.pl`

This table is the result of an analysis of primary EST sequence reads from the parasitic nematode *Litomosoides sigmodontis*. The first column provides links to the sequences analysed. The next five columns present the results of five different BLAST searches, with each cell linked to the relevant, saved BLAST search output. In the first column are searches using BLASTN against *L. sigmodontis* sequences to identify if the gene has been identified before. In the second are searches against genomic sequence from *Brugia malayi*, a related nematode, using TBLASTN, while in the third are BLASTX searches against the nr protein database. In the last two columns are BLASTX searches against custom protein databases from “ecdysozoans” (nematodes, arthropods and allies; to identify genes from this superphylum) and bacteria (to identify possible endosymbiont genes). In each BLAST result cell, one can quickly review the bit score and E-value to identify hits. Thus there are no hits to bacteria, but all five sequences have matches in the *B. malayi* sequence dataset.

### Figure 4: NEMBASE: a relational database for EST datasets

This is a screen snapshot of one entry in a database of nematode ESTs, NEMBASE (21). Each cluster of ESTs has been annotated with information such as the number and source of ESTs that it includes, the consensus sequences and a set of pre-computed blast searches against selected databases. The page also has links out to additional analyses,

such as BLAST, an alignment of the ESTs, and putative protein predictions. The database is running in the background, using a relational database management software called postgresSQL, and the world wide web page is generated “on the fly” using the web-database interface language php.

#### 4. Notes

**1** We have described an informatic solution that relies on the open source LINUX operating system which is installable on most modern PC processors. LINUX includes (several excellent) graphical user interfaces (point-and-click) and there are many free programs available for standard computing needs such as word processing, databasing and presentation. It is also possible to run much of the suggested software on non-LINUX platforms (Windows, MacOSX, UNIX flavours) but we have highlighted the LINUX platform because it has become the environment of choice for many bioinformatics researchers and thus is well supported and rapidly growing in diversity.

**2** As the LINUX machine will be networked, you may wish to discuss the versions of LINUX supported by your local computing resource people, and take their advice as to issues of machine purchase, adaptation, and security.

**3** There are many commercial software solutions to the issues of analysis of ESTs.

**4** In LINUX, the PATH environmental variable describes a list of directories which are automatically searched when you try to run a program (programs are also termed executables) from a command line prompt. If the executable is located in a directory not listed in your PATH then it can only be run by including the full directory path of the program. For example if you have the executable "blastall" in a directory called "/usr/ncbi/bin" and the directory "/usr/ncbi/bin" is not in your PATH, then in order to run that executable you would have to type:

```
/usr/ncbi/bin/blastall
```

at the command line prompt. With "/usr/ncbi/bin" in your PATH, this simplifies to:

```
blastall
```

The 'PATH' variable can be updated by editing the ".cshrc" or ".bashrc" file in your home directory. Which file is present will depend on the login shell that you use (see the LINUX manual pages on "bash" or "tcsh" for more information). In this paper we give the full path to the executable in the examples. In this chapter we assume you have installed executables and scripts in the places recommended in section 2.2.

**5** The **trace2dbest** package is undergoing continual development. Hence the operation of future releases may differ from that outlined here. The user is therefore asked to review the documentation which is bundled with each release (normally available as a menu option from the trace2dbest.pl executable).

**6** The standard format for output of data from a sequencing instrument is the *scf* or standard chromatographic format. All the major instruments can output in this format.

**7** Indeed, Phil Green's phred is now at the heart of many commercial solutions to the base calling problem, and "phred scores" is a standard piece of jargon in genome sequencing world wide.

**8** Naming conventions. It is vital that you develop a robust naming protocol for your clones and sequences. We suggest the following:

(a) A unique, two letter species code (thus "Eg" for *Echinococcus granulosus*);

(b) A simple, unique, library identifier code (thus "psc" for "protoscolex, or "ad1" for "adult, version 1");

(c) The unique microtitre plate address for the clone (thus "01A03" for plate 01, column A, row 03);

(d) Letters indicating the sequencing primer used (thus "T7" or "M13F").

These can be combined, most usefully with a "\_" (underscore) separating the different data fields: for example, a sequence derived using the T7 primer from clone A07 from plate 13 of an *Echinococcus granulosus* adult library (first version) could be named:

Eg\_ad1\_13A01\_T7

This format facilitates the use of perl scripts to extract from a sequence name relevant information, such as species, library, etc., without having to use a complex look-up table to attribute sequences to sources. The software we have used assumes that the above sequence naming scheme has been adopted. For example, trace2dbest assumes that the first two sections ("Eg\_ad1" in the above example) uniquely identify a cDNA library and the sequences derived from it. If you use a different scheme, it will be necessary to edit the perl scripts to match the patterns of names you use.

**9** FASTA format is very simple: each sequence definition line starts with a ">" and is followed by free text. After the next line return, all the characters are expected to be part of the sequence, until the next line beginning with a ">" or the end of the file is reached. The sequence itself can be in lines of any length and can spread over many lines separated by line returns. For example here are some sequences from the platyhelminth *Echinococcus granulosus*. Note that the FASTA ">" header line has been used to store additional information as well as the sequence name:

```
>gi|28395298|gb|AY187811.1| Echinococcus granulosus Hox5 mRNA, partial cds
GAGCTGGAGAAGGAGTTCCATTTCAACCGGTACCTCACGCGTCGGCGGAGGATAGAGATAGCCCACGCGC
TTTGCCATCTGAGCGACAGATCAAAATCTGGTTCCAAAACAGCCG
>gi|28395296|gb|AY187810.1| Echinococcus granulosus Hox3 mRNA, partial cds
GAGTTTGAGAAGGAGTTCCACTTTAACAGGTACCTGTGCCGCCCGCGGCGCGTTCGAGATAGCCAACCTCC
TGAATCTCACCGAGCGCCAAATAAAGATCTGGTTCCAAAACCGCCG
>gi|28268125|gb|CB219933.1|CB219933 EgP 38D7 signal sequence trap (SST)
GCGATACAATTAATAAAGGGAATAGAGTGAACGTTTCGGCCGGTTGATTTACAGCTGACCGCAGTCAGTAA
TCAGTACTTCTTGGGAAGTCTTGCCACTACTGCTACCGACGGCCTCCATATCCTTAACAACATCTTCGCC
ACTTTCTACCTCACCAAAGACAACATGCTTCCCATCAAGCCAGCTGGTGACGGCGGTAGTGATGAAGAAT
TGCGAGCCATTGGTGTCTTACCCGCATTTCGCCATCGAGAGCATCATCGGCTTGCTGTGCTTGTGATTGA
AATTTTCATCCTCAAATTTGCTCCCGTATATGCTCTTGCCACCGGTACCATTCCCGGCAGTAAAATCACC
ACCTTGGCACATAAAAACCGGA
```

**10** The taxonomy ID remains constant for each taxon in the NCBI database, so you can use this number to recall the sequence set by adding “txid####[Organism]” to your query in the search bar on the NCBI home page.

**11** We suggest that for most parasites, it makes most sense to download the nucleotide dataset, as these datasets will encompass the highest diversity of genes identified. For model organisms and organisms, such as *Plasmodium falciparum*, that have had their whole genomes sequenced and annotated, it is more efficient in terms of search time to download the protein dataset. However, if you are working on a close relative of a fully sequenced organism, you should also search the nucleotide sequence of the genome as your ESTs may correspond to genes missed in the annotation of the fully sequenced species. For the remainder of the sequenced biosphere, the nr protein dataset is the only one feasibly accessible to local search.

**12** It is not necessary to download the full nr protein dataset every time you update it, as GenBank provides an “update” facility. See <ftp://ftp.ncbi.nih.gov/blast/db/README> for instructions.

**13** A very useful tutorial on the use of BLAST, the sorts of parameter values that should and can be used, and the interpretation of BLAST outputs is available at the NCBI web site: <http://www.ncbi.nlm.nih.gov/Education/BLASTinfo/information3.html>. A detailed BLAST course by S.F. Altschul, the creator of BLAST is also available at <http://www.ncbi.nlm.nih.gov/BLAST/tutorial/Altschul-1.html>.

**14** A complete list of possible arguments available for the blastall command is revealed by typing “/usr/ncbi/bin/blastall -”. The current list is extensive (over 30 possible modifiers).

## References

1. Adams, M.D., Kelley, J.M., Gocayne, J.D., Dubnick, M., Polymeropoulos, M.H., Xiao, H., Merril, C.R., Wu, A., Olde, B., Moreno, R.F., *et al.* (1991) Complementary DNA sequencing: expressed sequence tags and the human genome project. *Science* 252, 1651-1656.
2. McCombie, W.R., Adams, M.D., Kelley, J.M., FitzGerald, M.G., Utterback, T.R., Khan, M., Dubnick, M., Kerlavage, A.R., Venter, J.C. and Fields, C. (1992) *Caenorhabditis elegans* expressed sequence tags identify gene families and potential disease gene homologues. *Nat Genet* 1, 124-131.
3. el-Sayed, N.M., Alarcon, C.M., Beck, J.C., Sheffield, V.C. and Donelson, J.E. (1995) cDNA expressed sequence tags of *Trypanosoma brucei rhodesiense* provide new insights into the biology of the parasite. *Mol Biochem Parasitol* 73, 75-90.
4. Wan, K.-L., Blackwell, J.M. and Ajioka, J.W. (1995) *Toxoplasma gondii* expressed sequence tags: insight into tachyzoite gene expression. *Mol Biochem Parasitol* 75, 179-186.
5. Blaxter, M.L., Raghavan, N., Ghosh, I., Guiliano, D., Lu, W., Williams, S.A., Slatko, B. and Scott, A.L. (1996) Genes expressed in *Brugia malayi* infective third stage larvae. *Mol Biochem Parasitol* 77, 77-96.
6. Ivens, A.C. and Blackwell, J.M. (1996) Unravelling the *Leishmania* genome. *Curr Opin Genet Dev* 6, 704-710.
7. Levick, M.P., Blackwell, J.M., Connor, V., Coulson, R.M.R., Miles, A., Smith, H.E., Wan, K.-L. and Ajioka, J.W. (1996) An expressed sequence tag analysis of a full length, spliced-leader cDNA library from *Leishmania major* promastigotes. *Mol Biochem Parasitol* 76, 345-348.
8. Ajioka, J.W., Boothroyd, J.C., Brunk, B.P., Hehl, A., Hillier, L., Manger, I.D., Marra, M., Overton, G.C., Roos, D.S., Wan, K.L., *et al.* (1998) Gene discovery by EST sequencing in *Toxoplasma gondii* reveals sequences restricted to the Apicomplexa. *Genome Res* 8, 18-28.

9. Djikeng, A., Agufa, C., Donelson, J.E. and Majiwa, P.A. (1998) Generation of expressed sequence tags as physical landmarks in the genome of *Trypanosoma brucei*. *Gene* 221, 93-106.
10. Manger, I.D., Hehl, A., Parmley, S., Sibley, L.D., Marra, M., Hillier, L., Waterston, R. and Boothroyd, J.C. (1998) Expressed sequence tag analysis of the bradyzoite stage of *Toxoplasma gondii*: identification of developmentally regulated genes. *Infect Immun* 66, 1632-1637.
11. Verdun, R.E., Di Paolo, N., Urmenyi, T.P., Rondinelli, E., Frasch, A.C. and Sanchez, D.O. (1998) Gene discovery through expressed sequence tag sequencing in *Trypanosoma cruzi*. *Infect Immun* 66, 5393-5398.
12. Ivens, A.C. and Blackwell, J.M. (1999) The *Leishmania* genome comes of age. *Parasitol Today* 15, 225-231.
13. Johnston, D.A., Blaxter, M.L., Degraeve, W.M., Foster, J., Ivens, A.C. and Melville, S.E. (1999) Genomics and the biology of parasites. *BioEssays* 21, 131-147.
14. Santos, T.M., Johnston, D.A., Azevedo, V., Ridgers, I.L., Martinez, M.F., Marotta, G.B., Santos, R.L., Fonseca, S.J., Ortega, J.M., Rabelo, E.M., *et al.* (1999) Analysis of the gene expression profile of *Schistosoma mansoni* cercariae using the expressed sequence tag approach. *Mol Biochem Parasitol* 103, 79-97.
15. Urmenyi, T.P., Bonaldo, M.F., Soares, M.B. and Rondinelli, E. (1999) Construction of a normalized cDNA library for the *Trypanosoma cruzi* genome project. *J Eukaryot Microbiol* 46, 542-544.
16. Williams, S.A. and Johnston, D.A. (1999) Helminth genome analysis: the current status of the filarial and schistosome genome projects. Filarial Genome Project. Schistosome Genome Project. *Parasitology* 118, S19-38.
17. Daub, J., Loukas, A., Pritchard, D.I. and Blaxter, M. (2000) A survey of genes expressed in adults of the human hookworm, *Necator americanus*. *Parasitology* 120, 171-184.
18. McCarter, J.P., Abad, J., Jones, J.T. and Bird, D.M. (2000) Rapid gene discovery in plant parasitic nematodes via expressed sequence tags. *Nematology* 2, 719-731.
19. Williams, S.A., Lizotte-Waniewski, M.R., Foster, J., Guiliano, D., Daub, J., Scott, A.L., Slatko, B. and Blaxter, M.L. (2000) The filarial genome project: analysis of the

nuclear, mitochondrial and endosymbiont genomes of *Brugia malayi*. *Int J Parasitol* 30, 411-419.

20. Degraeve, W.M., Melville, S., Ivens, A. and Aslett, M. (2001) Parasite genome initiatives. *Int J Parasitol* 31, 532-536.

21. Parkinson, J., Whitton, C., Guiliano, D., Daub, J. and Blaxter, M.L. (2001) 200,000 nematode ESTs on the net. *Trends Parasitol* 17, 394-396.

22. McCarter, J.P., Clifton, S.W., Bird, D.M. and Waterston, R.H. (2002) Nematode gene sequences, Update for June 2002. *J Nematol* 34, 71-74.

23. Parkinson, J., Guiliano, D. and Blaxter, M. (2002) Making sense of EST sequences by CLOBBing them. *BMC Bioinf* 3, 31.

24. Ewing, B. and Green, P. (1998) Base-calling of automated sequencer traces using phred. II. Error probabilities. *Genome Res* 8, 186-194.

25. Ewing, B., Hillier, L., Wendl, M.C. and Green, P. (1998) Base-calling of automated sequencer traces using phred. I. Accuracy assessment. *Genome Res* 8, 175-185.

26. Altschul, S.F., Gish, W., Miller, W., Myers, E.W. and Lipman, D.J. (1990) Basic local alignment search tool. *J Mol Biol* 215, 403-410.

27. Altschul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J., Zhang, Z., Miller, W. and Lipman, D.J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res* 25, 3389-3402.

28. Boguski, M.S., Lowe, T.M. and Tolstoshev, C.M. (1993) dbEST - database for "expressed sequence tags". *Nat Genet* 4, 332-333.

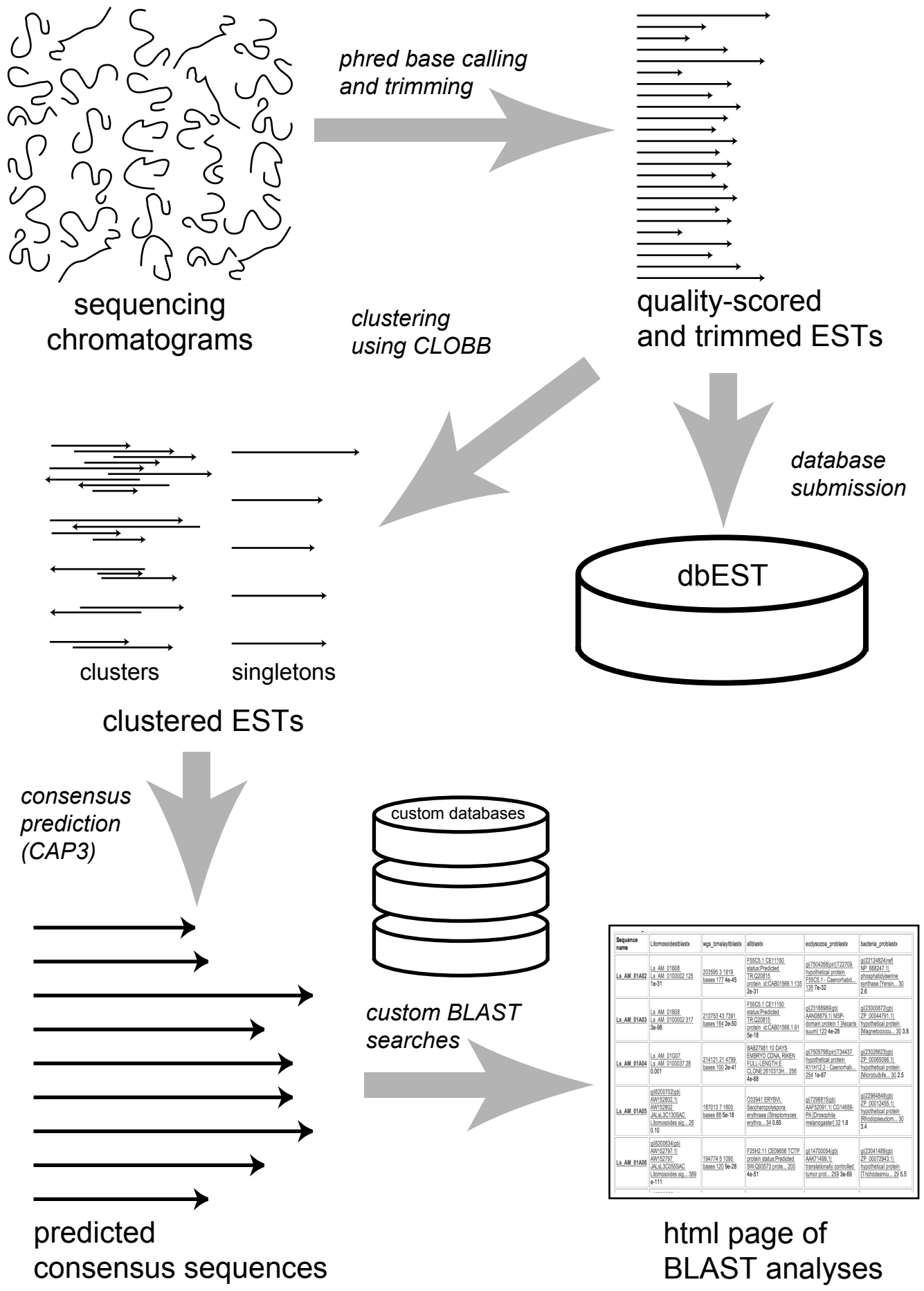
29. Christoffels, A., van Gelder, A., Greyling, G., Miller, R., Hide, T. and Hide, W. (2001) STACK: Sequence Tag Alignment and Consensus Knowledgebase. *Nucleic Acids Res* 29, 234-238.

30. Parsons, J.D., Brenner, S. and Bishop, M.J. (1992) Clustering cDNA sequences. *Comput Appl Biosci* 8, 461-466.

31. Parsons, J.D. (1995) Improved tools for DNA comparison and clustering. *Comput Appl Biosci* 11, 603-613.

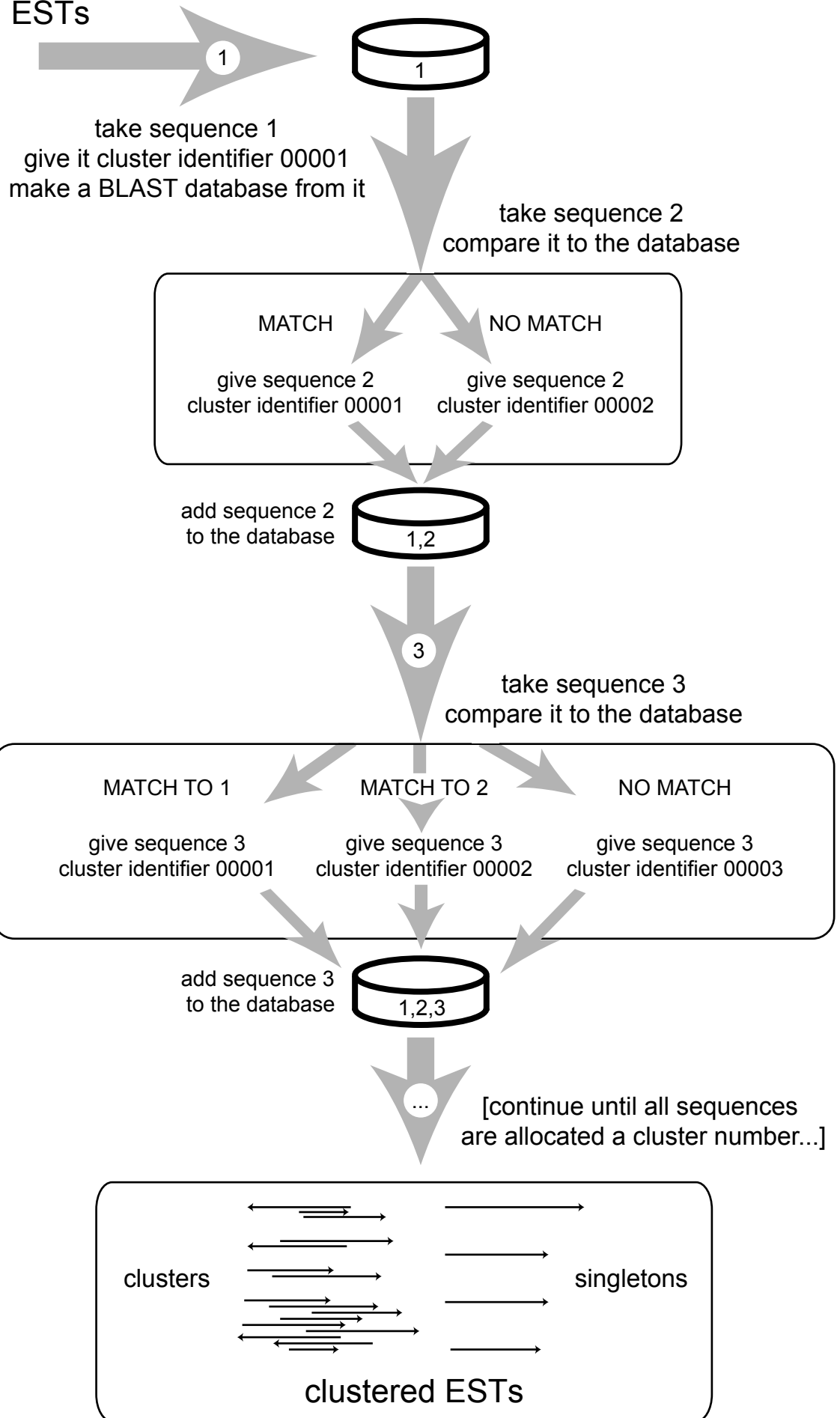
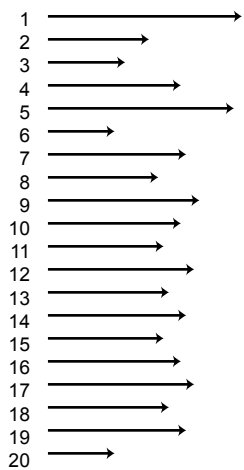
32. Gordon, D., Abajian, C. and Green, P. (1998) Consed: a graphical tool for sequence finishing. *Genome Res* 8, 195-202.

33. Huang, X. and Madan, A. (1999) CAP3: A DNA sequence assembly program. *Genome Res* 9, 868-877.
34. Parkinson, J. and Blaxter, M.L. (2002) SimiTri - visualising similarity relationships for large groups of sequences. *Bioinformatics* 19, 390-395.
35. Iseli, C., Jongeneel, C.V. and Bucher, P. (1999) In *Proc Int Conf Intell Syst Mol Biol*, pp. 138-148.
36. Fukunishi, Y. and Hayashizaki, Y. (2001) Amino acid translation program for full-length cDNA sequences with frameshift errors. *Physiol Genomics* 5, 81-87.
37. Hatzigeorgiou, A.G., Fiziev, P. and Reczko, M. (2001) DIANA-EST: a statistical analysis. *Bioinformatics* 17, 913-919.



Parkinson and Blaxter Figure 1

quality-scored  
and trimmed ESTs



Sequence name  
(in this case a *Litomosoides sigmodontis* nematode EST dataset)

Hit to database 1  
(cognate ESTs using BLASTN)

Hit to database 2  
(related nematode using TBLASTX)

Hit to database 3  
(nr protein using BLASTX)

Hit to database 4  
(proteins of arthropods and allies using BLASTX)

Hit to database 5  
(bacterial proteins using BLASTX)

Sequence name	Litomosoidestblastx	wgs_bmalayitblastx	allblastx	ectysozoa_problastx	bacteria_problastx
<b>Ls AM 01A02</b>	<u>Ls AM 01B08</u> <u>Ls AM 0100002 125</u> 1e-31	<u>203595 3 1819</u> bases 177 4e-45	<u>F55C5.1 CE11150</u> status:Predicted TR:Q20815 protein id:CAB01566.1 135 2e-31	<u>gi 7504268 pir T22709</u> <u>hypothetical protein</u> F55C5.1 - Caenorhabd... 135 7e-32	<u>gi 22124824 ref </u> <u>NP_668247.1 </u> <u>phosphatidylserine</u> <u>synthase [Yersin... 30</u> 2.6
<b>Ls AM 01A03</b>	<u>Ls AM 01B08</u> <u>Ls AM 0100002 317</u> 3e-98	<u>213753 43 7391</u> bases 164 2e-50	<u>F55C5.1 CE11150</u> status:Predicted TR:Q20815 protein id:CAB01566.1 91 5e-18	<u>gi 23168989 gb </u> <u>AAN08879.1  MSP-</u> <u>domain protein 1 [Ascaris</u> <u>suum] 123 4e-28</u>	<u>gi 23000872 gb </u> <u>ZP_00044791.1 </u> <u>hypothetical protein</u> <u>[Magnetococcu... 30 3.8</u>
<b>Ls AM 01A04</b>	<u>Ls AM 01G07</u> <u>Ls AM 0100037 28</u> 0.001	<u>214121 21 4799</u> bases 100 2e-41	<u>BAB27981 10 DAYS</u> <u>EMBRYO CDNA, RIKEN</u> <u>FULL-LENGTH E</u> <u>CLONE:2610313H... 256</u> 4e-68	<u>gi 7505798 pir T34437</u> <u>hypothetical protein</u> K11H12.2 - Caenorhab... 254 1e-67	<u>gi 23026623 gb </u> <u>ZP_00065096.1 </u> <u>hypothetical protein</u> <u>[Microbulbife... 30 2.5</u>
<b>Ls AM 01A05</b>	<u>gi 6200702 gb </u> <u>AW152802.1 </u> <u>AW152802</u> <u>JALsL3C130SAC</u> <u>Litomosoides sig... 26</u> 0.10	<u>187013 7 1600</u> bases 88 5e-18	<u>O33941 ERYBVI.</u> <u>Saccharopolyspora</u> <u>erythraea (Streptomyces</u> <u>erythra... 34 0.65</u>	<u>gi 7296815 gb </u> <u>AAF52091.1  CG14656-</u> <u>PA [Drosophila</u> <u>melanogaster] 32 1.8</u>	<u>gi 22964849 gb </u> <u>ZP_00012455.1 </u> <u>hypothetical protein</u> <u>[Rhodopseudom... 30</u> 3.4
<b>Ls AM 01A06</b>	<u>gi 6200634 gb </u> <u>AW152797.1 </u> <u>AW152797</u> <u>JALsL3C055SAC</u> <u>Litomosoides sig... 389</u> e-111	<u>194774 5 1090</u> bases 120 9e-28	<u>F25H2.11 CE09656 TCTP</u> protein status:Predicted SW:Q93573 prote... 200 4e-51	<u>gi 14700054 gb </u> <u>AAK71499.1 </u> <u>translationally controlled</u> <u>tumor prot... 259 3e-69</u>	<u>gi 23041489 gb </u> <u>ZP_00072943.1 </u> <u>hypothetical protein</u> <u>[Trichodesmiu... 29 5.5</u>

