CHAPTER 20

INFORMATICS FOR EFFICIENT EST-BASED GENE DISCOVERY IN NORMALIZED AND SUBTRACTED CDNA LIBRARIES

Todd E. Scheetz
University of Iowa
tscheetz@eng.uiowa.edu

Thomas L. Casavant
University of Iowa
tomc@eng.uiowa.edu

Beginning in 1997, large-scale efforts in EST-based gene discovery in Rat, Human, Mouse, and other species have been conducted at the University of Iowa. To date, these efforts have led to the sequencing of more than 400,000 ESTs accounting for more than 60,000 novel ESTs, including 40,000 previously undescribed genes.

This effort has been augmented by a set of custom informatics tools to gather, analyze, store, and retrieve the sequence data generated. The high rate of gene discovery associated with this work was primarily due to novel normalization and subtraction methodologies.[90] A critical aspect of this work is to periodically perform subtractive hybridizations of cDNA libraries against a set of previously sequenced clones from that library.

The informatics necessary for this effort consists first of examination of individual sequences to verify the presence of a 3' end, and to confirm the clone identity. Second, it is necessary to cluster the sequences to determine the current novelty of the library and to allow feedback to the cDNA library subtraction process to remove redundant clones and improve overall discovery efficiency. Finally, informatics is required for the submission of the EST sequences and associated annotation to both public and local databases.

In this chapter, the overall system of software is described, statistics concerning performance and throughput are given, and detailed methods of certain aspects of the pipeline are provided. In particular, our sequence editing, clustering, and clone verification methods are described in some detail. All originally produced softwares described here are available from our web server at `http://genome.uiowa.edu`, or by contacting `genome@eng.uiowa.edu`.

ORGANIZATION.

Section 1. We first provide an abridged biological experimentation background on EST-based gene discovery with cDNA libraries.

Section 2. Then we give an overview of a pipeline for EST sequence processing and annotation pipeline that we have developed for this purpose.

Section 3. The pipeline has five major components, *viz.* raw data gathering and archival, quality assessment and sequence editing, sequence annotation, novelty assessment, and submission to databases. These five components are described in detail in this section.

Section 4. Finally, we close with a discussion on the computational and storage resources required by various components of our pipeline.

1. EST-Based Gene Discovery

EST-based gene discovery is an efficient strategy in defining a gene index for an organism.[8] A gene index is a non-redundant collection of sequences, in which all sequences derived from the same gene transcript are grouped together. They provide a foundation to be annotated, and are an essential component in several analyses, including cross-species comparative analyses, and determination of unique sub-sequences—useful for SAGE, and for creating custom microarray chips. In addition, a non-redundant cDNA collection is useful in the creation of a cDNA-based microarray probe set.

The technology relies upon the sequencing of cDNA libraries, essentially partial DNA copies of mRNA transcripts. These cDNA fragments are typically sequenced on an ABI sequencer, resulting in a set of binary trace-files (chromatographs). The program phred[236] is used to extract the nucleotide sequence and per-base quality values. The phred quality values (a.k.a. phred values) are calculated as $q = -10 \times \log_{10} p$, where p is the estimated error probability for a given base. Thus, a phred value of 10 correlates with an error probability of 10%, a phred value of 20 with an error probability of 1%, and so on.

Expressed sequence tags (ESTs) are sequences derived from cDNA libraries, and have several common features, illustrated in Figure 1. These include a region of the vector sequence, and a cloning restriction site in front of the cloned sequence. In the case of a short cDNA insert, vector sequence may also be seen at the far end of the EST read. Because we utilize oligo-dT primed, directionally cloned cDNAs, our 3' ESTs typically begin with a polyT stretch, which correlates with the reverse complement of the polyadenylation (polyA) tail from the 3' end of mRNA transcripts. Similarly, the reverse complement of a polyadenylation signal is expected 11–30 bases down-stream from the detected tail.[144] To determine tissue of origin in a pooled library environment, a synthetic oligo tag is

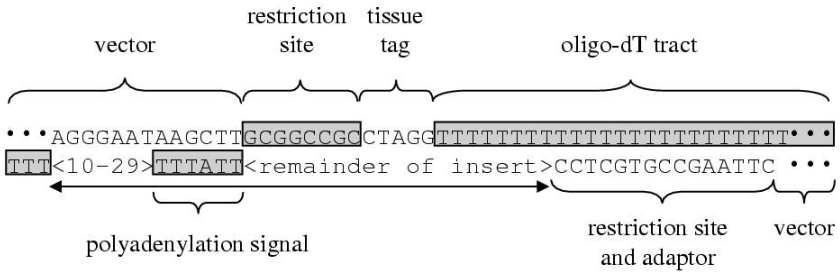


Fig. 1. Typical features found in ESTs derived from oligo-dT-primed, directionally cloned cDNA libraries.

inserted between the restriction site and oligo-dT during the creation of the cDNA libraries.

To maintain high rates of efficiency during the discovery process, we used the complementary approaches of cDNA library normalization and subtraction. [90] Both normalization and subtraction are used to reduce the prevalence of "undesired" clones from a cDNA library. In the context of gene discovery, undesired clones are either those we have seen before or are likely to see many times. Normalization addresses this issue by reducing the prevalence of the most abundant clones. Subtraction is a more focused procedure, reducing—ideally removing—the prevalence of a targeted set of clones.

2. Gene Discovery Pipeline Overview

We have developed an EST sequence processing and annotation pipeline, shown in Figure 2, to process 3' and 5' sequence data derived from cDNA clones. This pipeline supports all types of cDNA libraries—single-tissue or pooled, normalized or subtracted—from any species. The pooling of multiple tissues, all drawn from the same species, results in a more complex library and often enhances the effectiveness of serial subtraction, thereby improving the efficiency of the gene discovery process. However, the information regarding the tissue source of each clone is biologically interesting and useful. To collect this information from pooled multiple-tissue libraries, we use a method that "tags" each clone with an oligonucleotide sequence so that subsequent to generation of sequence data, the original tissue source can be determined. [279] While not directly useful as a definitive means for expression analysis, knowing the tissue an EST is derived from is a valuable resource. It should be emphasized that due to the subtraction processing, not all tissues in a pooled setting will reveal the presence of an EST for each

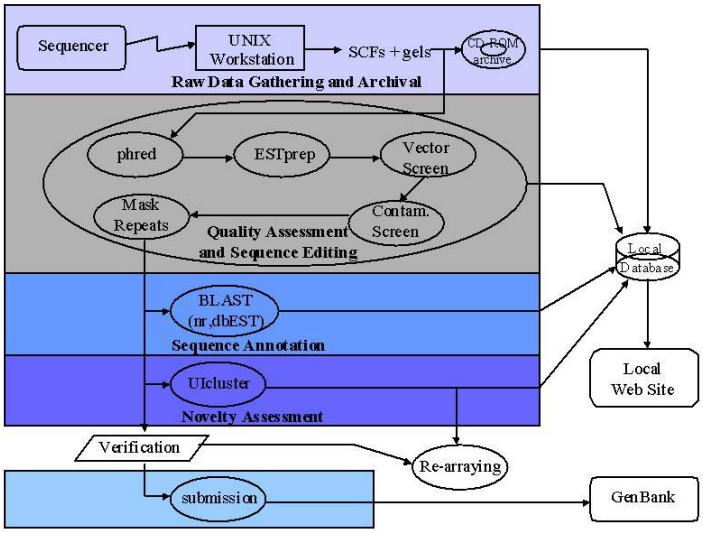


Fig. 2. Overview of the EST-based gene discovery pipeline.

mRNA message expressed by that tissue. Once a clone from a transcript is used in a subtraction, the prevalence of clones from that transcript will be greatly reduced.

The pipeline shown in Figure 2 can be viewed as consisting of five major components:

(1) Raw Data Gathering and Archival
(2) Quality Assessment and Sequence Editing (Feature Identification)
(3) Sequence Annotation
(4) Novelty Assessment
(5) Deposition or Submission to Local and Public Databases

The raw data gathering and archiving steps must assure efficient and secure communication between the data gathering nodes—nominally ABI sequencers—and the networked system of computers dedicated to the subsequent processing steps. This represents a challenge for a number of reasons, both technical and logistical. The quality assessment and sequence editing steps must assure that all sequence data processed by the remainder of the pipeline are of sufficient quality to be re-

liable. Both sequence integrity in terms of confidence of base calls, as well as confidence that the sequence is oriented properly and free of contamination, are supported. In the sequence annotation step, a number of sequence characteristics must be identified, labeled and stored for entry into a local database as well as submission to public databases. Well-known repetitive motifs are identified, and an initial BLAST examination against known nucleotide and amino acid databases is performed. Novelty assessment is based on a sequence similarity metric, and serves multiple purposes in this pipeline. Initially, all new sequences are clustered into the existing local "UniGene" set to calculate the current rate of novelty of the library being sequenced. As the final step in the pipeline, ESTs that are of sufficient quality, and are not contaminated are submitted, with annotation, to public databases such as dbEST and our local databases.

3. Pipeline Component Details

In this section, the software components of the large-scale gene discovery pipeline shown in Figure 2 are described. The pipeline is described from a functional perspective. Then each phase of the pipeline is described in detail. First, we describe the set of algorithms and software tools developed to initially process all sequences to confirm orientation, detect tissue tags, search for polyadenylation sequence and signal, and trim for overall sequence quality. Second, we describe a precise, high-performance tool for forming clusters of sequences which are likely to have been derived from the same gene or gene family. Finally, our method for verifying clone identity, and the informatics required to support it, is described.

The sequence-processing pipeline can be broken into the five broad phases listed in Section 2. In the Data Gathering and Archiving phase the "raw" chromatograph (SCF) files are collected from the ABI sequencers. The SCF files may be directly available (*e.g.*, from ABI 3700 sequencers), or obtained through tracking and extraction from a gel image (*e.g.*, from an ABI 377 sequencer). The naming of clones and sequences follows a standardized nomenclature, described in detail in Subsection 3.1. The sequence names are imported directly into the ABI sequencing software from sample sheets automatically generated via a web-based interface.

In the Quality Assessment and Feature Editing phase, phred[236] is used to obtain the base-calls and per-base quality values directly from the SCF files. These sequence and quality files are then processed by ESTprep.[751] The ESTprep application identifies several categories of features, including the region of high-quality sequence, commonly found in EST sequences. A complete description of the processing by ESTprep is covered in Subsection 3.2.

After processing with ESTprep, the sequences and their associated quality files are trimmed according to the identified regions of high-quality sequences. The trimmed sequence is then assessed for contaminating or repetitive sequences using RepeatMasker (`http://ftp.genome.washington.edu/RM/RepeatMasker.html`). Contaminating sequences fall into three categories—mitochondrial, bacterial (*E. coli*), and vector sequences. If significant amounts of contamination are found, the sequence is removed from the pipeline. Likewise, if a hit to the vector sequence is found only at the 3'-most end of the sequence, the matching portion is removed rather than discarding the whole sequence. Such EST sequences are derived from short cDNA clones, where the EST sequence "reads through" the cDNA insert and into the vector beyond. RepeatMasker is also used to mask repetitive and low complexity elements within the sequences. Here, the set of species-specific repetitive sequences from RepBase[404] is used as the set of sequences that should be masked. The process of masking takes bases identified as repetitive or low complexity, and replaces them with N's.

Sequences that reach this part of the sequence processing pipeline are of sufficient quality, length, not contaminated, and have had any repeats masked. They enter the Annotation phase. These sequences are blasted[24] against the non-redundant nucleotide database from GenBank and the appropriate species' sequences from dbEST,[87] as an initial annotation.

The masked sequences are also used in the Novelty Assessment phase, which utilizes the UIcluster program,[836] and is fully described in Subsection 3.4. The process of clustering the EST sequences is useful both for creating a non-redundant set of sequences and in assessing a library's gene discovery rate. The discovery rate of a library provides critical feedback to the library creation group for determining when normalization and subtraction processes should be performed.

The final phase is that of Sequence Submission. All sequences that have sufficient quality and are not contaminated are promptly submitted to the dbEST division of GenBank according to the "bulk email" format. This information includes the trimmed sequence with annotation describing some of the identified features—detection of polyA tail and polyA signal, any repetitive elements, library of origin, and tissue of origin. A complete snapshot of all the data from the sequence processing pipeline is loaded into our local database, and is available from `http://genome.uiowa.edu`. Our local database also stores comprehensive annotation for all locally generated ESTs, including expression, clustering and mapping information.

3.1. *Raw Data Gathering and Archival*

The first step in this stage is the generation of sample sheets to be loaded into the ABI sequencing software. This is an important first step in ensuring that the sequences are assigned names that conform to our naming structure described below. The sample sheets also store data on the technician, PCR block, and plate type—96- or 384-well—for each sequencing run. After the chromatograph files (SCFs) have been generated, they are transferred to a central drop-point. The drop-point is implemented on a Linux workstation which is exporting disk partitions to both Macintosh and Windows using netatalk (`http://netatalk.org`) and Samba (`http://samba.org`) respectively.

Next, the chromatograph files are copied to the appropriate project directory onto a local RAID disk array. Each project has a separate directory hierarchy divided into (minimally) 3' and 5' sequence directories. Within the direction-specific directories a plate of sequences is stored within a directory hierarchy consisting of run date, machine name, and plate name. This allows us to track progress by date, and to identify systematic problems with a specific sequencer or capillary. The ABI-derived chromatograph files and the gel image, if run on a 377, are archived onto CD or DVD-ROM for long-term storage.

Because the EST processing pipeline works on large numbers of cDNA libraries, a clone naming convention was designed to quickly identify what library a given sequence or clone was derived from. The naming format is essentially a set of eight values separated by dashes. The first value denotes the originating institution. For all of our libraries, the value "UI" is used, indicating that those clones came from the University of Iowa. The second and third values denote the project and library codes. The plate, row, and column that specify a well are the fourth, fifth, and sixth values. Finally, the seventh and eighth values are for the replication number and replicating institution. These are important for tracking sets of clones that have been distributed, such as re-arrayed sets sent to Research Genetics. The replication number of "0" is reserved for the original "master" plates. So the clone name of "UI-R-A0-ad-e-04-0-UI" refers to a clone from the UI rat project (project code "R") library "A0". The clone is located at column "e", row 4 of plate "ad". The final two data points tell us that the clone is from the original master plate arrayed at the University of Iowa. Further information on our clone naming protocol can be found at `http://ratEST.eng.uiowa.edu/localdocs/naming.html`.

3.2. *Sequence Assessment and Sequence Editing*

As mentioned earlier, this stage both verifies the quality of the EST sequences, and identifies common features. The quality is assessed based upon the per-base values assigned by phred and detection of contaminated ESTs. The features to be identified are dependent on the EST end sequenced (3' or 5') and on the specifics of the cloning procedure.

Phred[236] is used to extract the initial base calls and per-base quality values from the "raw" SCFs. New chromatograph files are also generated from the ABI-based files. This ensures that the base-calls embedded within the SCF are consistent with those determined by phred. An additional benefit is that the SCF files generated by phred are also significantly smaller than those generated by the ABI sequencing software.

Next, the ESTprep[750] program is used to perform an initial quality assessment, and to identify features common to ESTs. The fundamental procedure underlying ESTprep is to first identify candidate sites representing the expected cloning site. These sites are then validated by searching for the expected vector sequence adjacent to the predicted restriction site. Based upon the identified location of the cloning site, other features may be identified if applicable, including library tag, polyA tail, and polyA signal. All features to be detected are configurable in the set of features to be identified, specifics of the features and in the allowable number and types of errors. For clarity, a diagram of these features is shown in Figure 3.

The following description of ESTprep is provided using a set of default parameters to provide context. Most of these parameters are configurable through modification of a configuration file shown in Figure 4. The first stage within ESTprep is an initial quality assessment. During this stage the sequence quality is verified, and any low-quality stretches of sequence at the beginning are removed to avoid detection of spurious features. If the average sequence quality over the first 200 bases is less than 20, the sequence is removed from the pipeline. Similarly, a leading stretch of sequence with fewer than 8 out of 20 bases greater than 20 will be removed if present.

Next, the program attempts to identify the restriction site used during the cloning process. The strategy used during the restriction site identification is to first identify high-quality restriction site candidates and to validate those candidates by looking for the expected vector sequence adjacent to the site. The quality of the candidate restriction sites is gauged by the number of errors, or edit distance,[309] away from the "correct" restriction site.

As mentioned above, the first feature identified is the restriction site. For our pipeline, we use an eight base recognition site. If the actual recognition site used

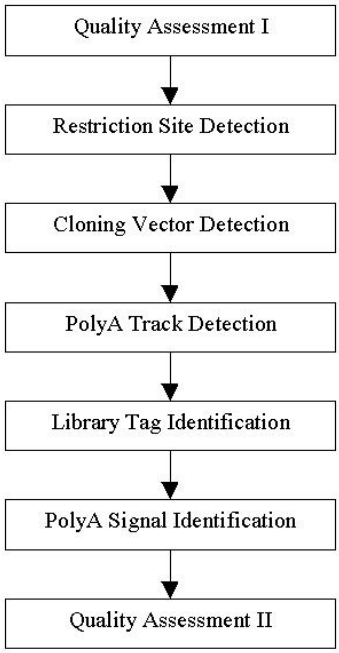


Fig. 3. ESTprep processing stages.

in cloning is not eight bases long, a "synthetic" restriction site is constructed using the last eight bases of the vector sequence (including the restriction site) prior to the cloned cDNA insert. In Figure 1, this sequence is the recognition sequence of the NotI restriction enzyme (GCGGCCGC). When the recognition site is less than eight bases, a synthetic restriction site is constructed. The adjacent portion from the vector/polylinker are prepended to make the synthetic restriction site. The presence of vector sequence leading up to the restriction site is used to confirm that this site corresponds to the bona fide cloning vector site. This is necessary because of errors inherent in the sequencing process. Sequence errors may occur in the restriction site, making it unrecognizable, and alternatively they may alter another region such that it then resembles the restriction site.

Feedback based upon empirical assessment by the sequencing group was used in refining the quality assessment parameters. ESTprep allows rejection of sequences based upon average sequence quality, number of phred 20 bases, and polyA tail length. The current defaults used in our EST sequence quality assessment are shown in Figure 4 and described in detail below.

For 5' EST sequences, detection of a restriction site with leading vector, and the assessment of quality is a complete analysis. However, for 3' ESTs several other features are assessed—polyA tail, polyA signal, and library tag. The polyA tail and polyA signal are representative of the polyA tail and signal in the mRNAs from which the cDNA library was derived. Because of the procedures used in library construction the polyA/T stretches in the cDNA library are tightly controlled. Thus, rather than a detected polyA tract of a few hundred nucleotides, the polyA tract of mRNAs from well constructed cDNA libraries averages approximately 26 nt in length. The expected length is a user-specified parameter in the prep.params file.

After the restriction site has been located, a sequence is parsed downstream to find the first nucleotide of the polyA tail - typically T in 3' ESTs. The algorithm for detecting the starting and ending positions of the polyA tail is surprisingly complex, involving a number of carefully crafted heuristics. A brief summary of this algorithm is provided here. Starting from the ending position of the putative library tag in the input sequence, a floating-point values array of T-density is constructed. This density array is parsed from a position beyond the maximum length (user-specified) of the polyA tail. This parsing proceeds backwards until a point at which the density exceeds a specified threshold. In this way, the right-most extent of the region richest in 'Ts' is located. If the T-rich region does not end with a T, then the ending point is recursively retracted by one, providing a more accurate identification of the polyA tail extent. Finally, if the polyA tail identified is not of sufficient length, the detection process is repeated, beginning one base right of the original starting base. Once the right-most extent of the polyA tail has been located, the process is resumed to the left to attempt to identify the left extend of the polyA tail. A lower density threshold (94% of original) is used so as not to truncate longer polyA tails, while mis-identifying smaller portions of polyA stretch immediately adjacent to the restriction site. If a valid-length of polyA tail can still not be identified, the sequence will not have a library tag associated with it.

If a polyA tail is identified, the search begins for a corresponding polyadenylation signal. Specification of custom sets of polyadenylation signals is supported. Typically, the two canonical polyadenylation signals (AAUAAA, AUUAAA) would be provided by the user, and a second set may be specified for other alternative signals. The default configuration is to accept the pair of canonical signals, and in addition, the alternative signals identified in Beaudoing *et al.*[72] Polyadenylation signals must be found within 11–30 nucleotides from the right-most end of an identified polyA tail in 3' ESTs.[144]

The final feature identified by ESTprep is the library tag. This is typically a

```
Sequence_Direction: FORWARD
Echo_Sequence: 2
Quality-Params: 10 8 20 100
Restriction-Site-Tag: 200 2 1 2 GCGGCCGC
Cloning-Vector: 6 GCCAAGCTAAAATAACCTCACTAAAGGGAATAAGCTT
LibTags: 1 1 1
CAGCC
rat-eye
polyAtail: T 10 .95 20 18 18 .65
polyAsignal: 11 30 2 0 0 0 TTTATT TTTAAT
polyAsignal-alt: 14 TTTACT TTTATA TTTATG TTTATC TTTAGT TTTAAA
TTTCTT CTTTTT TGTTTT AGCCCC TATATT TGTATT TCTATT TTCATT
```

Fig. 4. An example parameter file for ESTprep.

stretch of 4–10 nucleotides inserted during library construction to aid in identifying the tissue of origin for a given EST sequence. For an in-depth description of the processes used in the library tag construction and detection, see Gavin *et al.* [279] As shown in Figure 1, the library tag lies between the restriction site and the polyA tail. Without identification of both the restriction site and polyA tail, identification of the library tag is not attempted. Library tag identification is performed by constructing a substring of sequence between the restriction site and the polyA tail. This sequence is then used in a local sequence alignment versus all possible library tags, as specified in the prep.params configuration file; see Figure 4.

The final stage in ESTprep processing is a second quality assessment. While fully user-configurable in prep.params, typically, a 20 base-pair sliding window is applied looking for the first region which contains 8 bases with phred quality values less than 10. The sequence trim location for the right-most end of the sequence is then the start of the first window meeting these criteria. Typically, if the resulting trimmed sequence would be less than 100 base-pairs long, the sequence is rejected from the pipeline, as dbEST does not accept sequences less that 100 bp in length. If the average quality of the trimmed sequence is less than 20 or if the fraction of bases with quality greater than 20 is too low (e.g., less than 53%) the sequence is similarly rejected from the pipeline.

The output of ESTprep is a summary file listing each feature identified. Consider a sequence file named “foo”. The resulting summary file would be named “foo.prpsmry” (see Figure 6). A more verbose, and human readable descriptive transcript of the resulting analysis is also printed to STDOUT, which is typically redirected to a file for later analysis, or for posting to a web page for project monitoring.

The format of ESTprep’s parameter file is shown in Figure 4. The Se-

```
>UI-R-CV0-bro-h-12-0-UI.s1 608 0 608 ABI
TGACGGCCAGTGCCAAGCTAAAATTAACCCTCACTAAAGGGAATAAGCTT
GCGGCCGCCAGCCTTTTTTTTTTTTTTTTTTTTTTTTAACAATATTTGT
ACCGTTTTTATTTGTAAAAATAACCATCTGAATGCATGTCCATCGTATGT
TACAGGTAAGTACTTCATTGCATATTAGAGCACTCAGTAGTTGGGAAAGT
ATTAACCTGTGCTTGGGAAGATTCACACTGGTCCAAAGTCCTTCACTGAA
CCCAATGCCATTTTTCCTCATTTTTTACACTCAGGACGTTTACCAAAGTC
ACTCAACTCCAATCTACATCTTAAAATTACAGACGAAAAAATCCCACTGA
AATCATCACATAATAGTTTATGCTGTTACAAACACGTTTTACAAAATGTT
ACACTGGCATAAGATGTGTATCACCGTGCTTTGCAAGGATATATTGCACA
ATGCTGAAGCGTGGTCTGGAGACAAAACTGCTAAACAAAAGATTCTCCCC
TCGAGTCCTCAGATCACCGGGTAAGAAACACACAGAATTTCATCTTACAA
ACACGCTCAGATTGTCACTATCTTAAAACGCTGTCCTCCTATAATACTGA
CCTATGGG
```

Fig. 5. An example 3' EST sequence.

```
CLONENAME: seq/UI-R-CV0-bro-h-12-0-UI.s1.seq
RESTRSITEFOUND: TRUE
VECTORFOUND: TRUE
VECTORLENGTH: 33
POLYATAILFOUND: TRUE
POLYATAILLENGTH: 25
LIBTAGFOUND: TRUE
LIBTAG: CAGCC rat-eye
POLYASIGNALFOUND: TRUE
POLYASIGNAL: TTTATT
TRIMLOC: 71 451
GOODQUALITY: 25.36
QUAL_FILTERS: 25.36 39.84
STATUS: GO
```

Fig. 6. A Prep summary file resulting from running ESTprep on the sequence in Figure 5 using the params list in Figure 4.

quence_Direction field is used to denote which direction the EST sequence is expected to be in. Valid possible values are FORWARD and REVERSE. When processing FORWARD sequences, 3' relevant features are identified, including polyadenylation tail and signal, and library tag. The Echo_Sequence field controls the verbosity of output from ESTprep. Next are the parameters used in quality assessment. These parameters are specified by four integer numbers following the Quality-Params field. The first three values (10 8 20, above) specify the parameters used in the quality-based trimming procedure. These values specify that the left trim-site will be at the first position where there are 8 positions within a win-

dow of 20 that are less than phred quality of 10. The fourth value specifies the minimum number of bases that must remain after trimming. The example in Figure 4 specifies that a minimum of 100 nucleotides are required after trimming for the sequence to satisfy the quality criteria.

The Restriction-Site-Tag field is used to specify a recognition sequence identifying the restriction recognition site. In cases where the actual recognition site is less than eight bases long, the right-most eight base sequence is used, which may include a few bases of the cloning vector. Additional upstream vector sequence is provided with the Cloning-Vector field. Identification of the expected vector sequence upstream of an identified restriction site is used to validate the restriction site.

The LibTags field is used to define the set of expected library tags. The definition is divided onto three separate lines. The first line specifies the number of tags in the set, along with specification of allowable error properties. In the example above, the values "1 1 1" denote the fact that this library contains a single valid library tag, and that the detection scheme may allow up to one substitution or one insertion/deletion error when identifying the library tag. The second and third lines are used to specify the valid library tags and their correlated source tissues.

The polyAtail field is used in identifying the polyadenylation tail from 3' ESTs. Seven parameters are specified. The first is the character for which the tail will be identified. In the case of typical 3' ESTs this is T, the complement of A. The next four values specify limiting values on a subsequence in order for it to be identified as a polyA tail. The first of these (10) specifies a minimum polyA tail length of 10. The polyA tail is then extended from an original exact match of 10 T's. The next two parameters require that in the event that the expected tail sequence does not begin with a series of 10 consecutive T's, that the detected tail be 95% T over the first 20 bases. The 95% threshold also defines the criteria for determining the end of the polyA tail. As the tail is extended the relative percent T is calculated. The end of the tail is determined as the last position prior to the percent T $< 95\%$ that is a T. The fifth parameter specifies the maximum number of bases from the identified tail sequence that should be retained in the trimmed sequence. The final two values (18 and .65) are used to identify a probable polyA tail when the more stringent 95% parameter fails to identify exact tail boundaries. This is often the result of regions of low-quality sequence, or internal priming events.

The polyAsignal field is used to specify the set of potential polyA signals to be identified, and parameters to limit the search. The first two values limit the distance within which a polyA signal may be detected from an identified polyA tail. The values of 11 and 30 are used in the example above, as published in

Chen *et al.*[144] Therefore, a valid polyA signal must begin from 11 to 30 bases from the identified end of a polyA tail. The third value specifies the number of polyA signals in the set to be searched, in this case two. The next three numbers specify the allowable errors—missense, insertion, and deletion. By default these are set to zero when searching for a specific set of alternative polyA signals. The remaining words are the canonical polyA signal candidates.

The polyAsignal-alt field is similar to the polyAsignal field, but used to differentiate between canonical polyA signals (*e.g.*, TTTATT and TTTAAT) and alternative polyA signals. The first value specifies the number of alternative signals that may be identified. The remaining values specify the valid alternative polyA signal candidates. No errors are allowed during the identification of alternative polyA signals.

The contamination detection and repeat masking phase of the EST processing pipeline utilizes the RepeatMasker package as the core sequence similarity utility. The benefit of using RepeatMasker over an alternative alignment algorithm is its sensitivity in detecting distantly related sequences. This is especially important when masking repetitive elements, which may have diverged significantly from the canonical form. The cost of using RepeatMasker is measured in processing time. RepeatMasker utilizes cross_match to perform its sequence alignments which is an efficient implementation of the Smith-Waterman-Gotoh algorithm.[300,780]

The computational overhead inherent in using RepeatMasker is only significant when screening versus a large database of sequences, such as the bacterial genome. In this case, a different alignment algorithm could have been selected for the different screening needs (bacterial, mitochondrial, vector, repeats). The identification of repetitive elements requires a sensitive alignment algorithm. In contrast, assessing for contaminants could be performed with a less sensitive algorithm, such as BLAST. However, the decision to use a single package simplified the design of the programs that execute and parse the resulting outputs.

RepBase[404] is used as the baseline database of repetitive elements to be masked for a given species. Any hit to a repetitive element is masked, meaning that the responsible sub-sequence is replaced base-for-base with a string of N's.

When assessing for bacterial or mitochondrial contamination, a sequence is considered contaminated if more than 85% of the sequence matches the bacterial or mitochondrial database. Assessing for vector contamination is slightly more complex. Two classes of contamination from vector sequences are possible. The first is complete vector contamination, *i.e.*, there was no insert. For this case, a criteria of 85% is used, just as in the assessment of bacterial and mitochondrial contamination. The other potential vector contaminant is that of a sequence de-

rived from a short cDNA insert. In this case, only the end of the EST sequence is expected to match vector sequence. The remaining sequence is valid cDNA sequence. Because the amount of vector sequence may be limited, a low detection threshold is used. To limit the number of false-positive matches, an additional requirement is applied, requiring the detected vector sequence to extend to within 10 bases of the end of the trimmed sequence. When screening for vector contamination, it is important to utilize the sequence of the specific vector used in the cDNA library. This ensures that weak hits at the end of a short cDNA insert are detected.

Sequences detected with complete contamination are removed from the pipeline. For sequences with detected trailing vector sequence, the subsequence matching the vector is trimmed. Similar trimming is also performed on the quality file to maintain synchronicity with the trim file.

A final quality assessment utilizes a sample-sequence based verification. This is performed by rearraying eight clones from each plate of clones arrayed for sequencing. These sets of eight clones are arrayed into a new plate, referred to as a verification plates. These verification sequences are then compared to the original sequences. In the event that the two sequences (original and verification) do not match, the clone is marked.

The verification procedure provides valuable feedback on the consistency of clone-sequence correlation. As an example, the verification data has been used to identify a problematic plate that was re-sequenced. The verification data is also useful in prioritizing candidates for inclusion in non-redundant sets. Clones with positive verification results are preferentially selected, while those with negative verification results are selected against.

3.3. *Annotation*

An initial annotation of the ESTs is a useful set of data for investigators using the ESTs. All uncontaminated, high-quality ESTs, as determined from the previous phase, are blasted against the non-redundant nucleotide and amino acid databases from the National Center for Biotechnology Information (NCBI). When available, a database of species-specific ESTs is also blasted against to provide initial cross-references into UniGene clusters.

3.4. *Novelty Assessment*

The cDNA library normalization and subtraction techniques rely upon efficient novelty assessment to maximize discovery. A typical method used to assess novelty in cDNA libraries is based upon clustering the ESTs. The process of cluster-

ing partitions a set of input sequences based upon sequence similarity. This aids in assessment of library novelty and in the definition of non-redundant sets. The UIcluster clustering program[836] is used for our clustering analyses. This program can be run on a single CPU or in parallel, utilizing the MPI toolkit.[258] The fundamental strategy used in UIcluster is to add new sequences to an existing set of clusters one at a time. Each sequence is compared to a representative element from every cluster. If the sequence has significant similarity to a cluster's representative element, it is added to that cluster. In the event that the sequence has a minimal similarity to more than one representative sequence, it is added to the cluster it is most similar to, with stored annotation on which other clusters it matched. Masked EST sequences are used as the input to the clustering process. This restricts sequences from being grouped together based upon repetitive or low-complexity sequence.

To build the most accurate clustering possible, the available full-length mRNA sequences for the given species are added in to the clustering. Results from our Rat Gene Discovery and Mapping Project (T. Scheetz, in preparation; `http://ratEST.uiowa.edu`) indicate that incorporation of mRNA sequences merge clusters that would otherwise be disjoint for a significant fraction of the mRNAs. Multiple clusters may arise from a single transcript due to cDNA library artifact such as internal priming or restriction sites, or from alternative processing such as alternative polyadenylation or alternative splicing.

The default criteria used to determine minimal sequence similarity is an alignment of 38 out of 40 (95%) bases. Both misread and insertion/deletion errors are allowed. The minimal alignment is extended to the maximal amount allowed by sequence homology. Sequences may match in either orientation, *i.e.*, forward or reverse complemented. The resulting alignment is saved in the output clustering file, an example of which is shown in Figure 7.

The example shows a cluster of three sequences. The representative element (primary) is denoted with the "@P" tag, followed by the name of the representative element. This line is followed by the nucleotide sequence of the representative element. Other cluster elements (secondaries) are denoted with the "@S" tag. The @S tag is followed by annotation specifying the extent of match between the secondary and the representative element. The first two numbers are integers representing the start of alignment in the primary and secondary respectively. The third number is the length of the alignment, and the fourth is the percent identity of the alignment. The final tag denotes the direction of the match. This can be one of FORWARD, REVCOMP, or ORPHAN. The ORPHAN tag is used when a sequence no longer matches the current primary. Orphans can only occur when the *repick* option is used, causing re-evaluation of the primary element during cluster-

```
@P: UI-R-DO1-cml-n-03-0-UI.s1 0
TTTTTTTTTTTTTTTTTTGGTCAGGAAATTTTATTTGAACATTCTAAAGCAAGAATGCTTCAGATGTTAC
TTAAATGTCCCAGACAGGATTAACAAAATTAAATGTTTCTAAATTACAAATTTAGCTCCAGTAGGAGTTT
CATAAAAGAAGAAAACAACCCCCTCCCAAAAGAAGTATGACACACACATTCTGAAGAAACCCCAATGTTT
CATGCAATGGTAGGCAAGATGTAGAAGGCCACCCAAACCCATCTGTTTCTACACCAGTCATCACCCCGAA
GAGTCCTCCAGTCAATCTGTACATCCAAATGCATCCGGGAACCTACACCTACAAGACATTATTAATGTTA
TATACATTTATTGCCCCCCTTGGTTTTTTTAATAATTTCTTATGTAAAGCCTTCATTGAAACCCAAAAAA
AAAAAAAAAAGGATGTAAGACTAACTTGGGGGTAGGGAGGGGAAGATAATCACTTTAGACATTCAGTTAA
AATGTAAATTATCTAAATCTCCAAATGTTTAATAAAAACAAGCATCTTCTCCATTTAACACCTTGCTTGT
TAACTGTACAGTAAATTGTATTATAGAGAGTACATCTCTATTTTCATACTGTATCTTCTTTGGATGGAAT
TGAGAAAGCTGGTTAATTTTAAGATAAATAAATGAGATTGATCCAACTAAGATTAAGATGACAGCAGATA
TATTCCATGCAGAATTTAATAGTTTTTAATTTGT

@S: UI-R-DO1-cmm-n-05-0-UI.s1 18 18 704 100.000000 FORWARD
TTTTTTTTTTTTTTTTTTGGTCAGGAAATTTTATTTGAACATTCTAAAGCAAGAATGCTTCAGATGTTAC
TTAAATGTCCCAGACAGGATTAACAAAATTAAATGTTTCTAAATTACAAATTTAGCTCCAGTAGGAGTTT
CATAAAAGAAGAAAACAACCCCCTCCCAAAAGAAGTATGACACACACATTCTGAAGAAACCCCAATGTTT
CATGCAATGGTAGGCAAGATGTAGAAGGCCACCCAAACCCATCTGTTTCTACACCAGTCATCACCCCGAA
GAGTCCTCCAGTCAATCTGTACATCCAAATGCATCCGGGAACCTACACCTACAAGACATTATTAATGTTA
TATACATTTATTGCCCCCCTTGGTTTTTTTAATAATTTCTTATGTAAAGCCTTCATTGAAACCCAAAAAA
AAAAAAAAAGGATGTAAGACTAACTTGGGGGTAGGGAGGGGAAGATAATCACTTTAGACATTCAGTTAA
AATGTAAATTATCTAAATCTCCAAATGTTTAATAAAAACAAGCATCTTCTCCATTTAACACCTTGCTTGT
TAACTGTACAGTAAATTGTATTATAGAGAGTACATCTCTATTTTCATACTGTATCTTCTTTGGATGGAAT
TGAGAAAGCTGGTTAATTTTAAGATAAATAAATGAGATTGATCCAACTAAGATTAAGATGACAGCAGATA
TATTCCATGCAGAATTTAATAG

@S: RPLAH01TF 19 18 417 99.520384 FORWARD
TTTTTTTTTTTTTTTTTGGTCAGGAAATTTTATTTGAACATTCTAAAGCAAGAATGCTTCAGATGTTACT
TAAATGTCCCAGACAGGATTAACAAAATTAAATGTTTCTAAATTACAAATTTAGCTCCAGTAGGAGTTTC
ATAAAAGAAGAAAACAACCCCCTCCCAAAAGAAGTATGACACACACATTCTGAAGAAACCCCAATGTTTC
ATGCAATGGTAGGCAAGATGTAGAAGGCCACCCAAACCCATCTGTTTCTACACCAGTCATCACCCCGAAG
AGTCCTCCAGTCAATCTGTACATCCAAATGCATCCGGGAACCTACACCTACAAGACATTATTAATGTTAT
ATACATTTATTGCCCCCCTTGGATTTTTTAATAATTTCTTATGTAAAGCCTTCATTGAAACCCAAAAAAT
AAAAAAAAGGATGTAAGACTAACTT
```

Fig. 7. Excerpt from a UIcluster output.

ing. With this option, the longest constituent sequence is selected as the primary element.

To support efficient processing of 100,000's of ESTs several features have been integrated into UIcluster to accelerate the clustering process. Most significantly, incoming sequences are only compared to the primaries of each cluster, and a hashing scheme is used to efficiently perform an initial assessment of similarity. The hashing strategy computes a numeric value for each ξ-base subsequence, as shown in Equation 1. In this equation, K represents the size of the symbol alphabet—for nucleotide sequences, K is equal to 4. The variable Φ_i represents the ith letter in the subsequence. Thus each potential subsequence generates a unique hash value. Subsequences including a non-standard base call—*i.e.*, not A, C, G, or T—are not hashed.

$$H = \sum_{i=0}^{\xi-1} K^i \times \Phi_i \tag{1}$$

For the default ξ of 8, this yields 4^8 possible values based on the four-character DNA alphabet $\{$A, C, G or T$\}$. The presence of any other symbol results in an invalid hash. The hash values for each cluster representative are maintained in a global data structure, the Global Hash Table. Hash values are also created for each incoming sequence, and a complete sequence similarity assessment is only performed if at least one hash value is shared between the incoming sequence and a cluster representative. This filtering strategy greatly reduces the time to compute a clustering of EST sequences, but with a significant increase in the amount of memory required to store the Global Hash Table.

The hash size is a run-time configurable option. The larger the hash that is used, the more efficient the comparison becomes. There are two practical limiting factors on the size of hash that can be used. The first is architecture specific. On a 32-bit architecture it is inefficient to use a hash size of more than 16 bases, since $2^{32} = 4^{16}$. The second limitation is determined by the similarity required to cluster two sequences together. The less strict the alignment criteria is, the shorter the maximum viable hash size. This reflects the increased impact of a single-base error has on the hashes. Take as an example the default clustering criteria of 38 out of 40bp. If a hash size of 16 is used two errors could cause all 16 base hashes within a 40 bp region to not match their error-free counterparts. Equation 2 provides a guide for determining the maximum viable ξ for a given clustering criterion, expressed in terms of N identical bases in a subsequence of length M.

$$\xi = \left\lfloor \frac{M}{M - N + 1} \right\rfloor \tag{2}$$

Testing for cluster membership is evaluated first on clusters with the largest number of similar hash elements.

3.5. *Submission to Local and Public Databases*

Sequence, clone, and clustering data are deposited in a local database. The schema for this data is shown in Figure 8. The local database infrastructure also supports mapping information (not described). The sequence and clone records contain information on the sequence-specific features and annotation. The cluster information maintains the cluster assignments, local and UniGene, for each sequence. Additional cluster-specific information summarizing the features of the constituent sequences of each cluster is maintained in a separate table. Often, during the initial loading of data, certain information may not be available. The NCBI accession and GI numbers, and UniGene cluster membership are chief among these, as they are determined by remote sites (NCBI). To maintain a comprehensive data set, programs are available to update those data as the information becomes available.

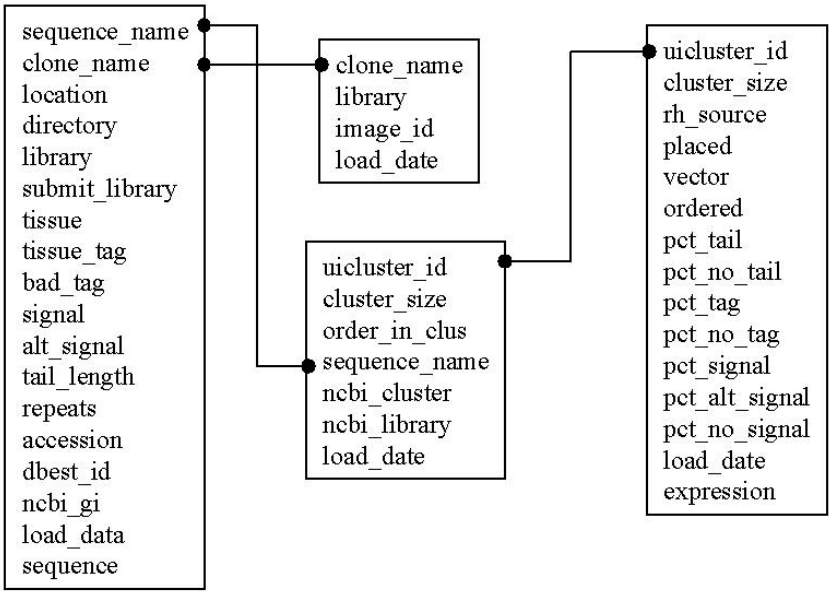
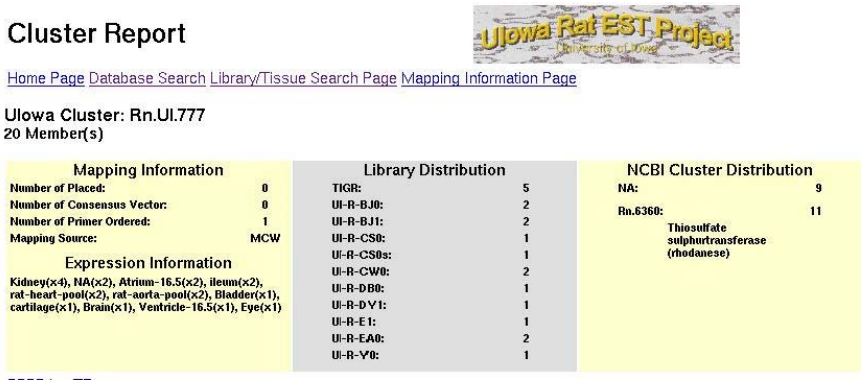


Fig. 8. EST-specifc database schema.

Several Perl-based web interfaces have been implemented to allow access to the stored information via the Internet. Specifically, a search tool is available, as well as report generation interfaces for clone-, sequence-, and cluster-specific information. Examples of these interfaces are provided in the following figures. Figure 9 shows the top of the Cluster Report interface. A summary of the cluster is provided at the top of the report (shown), with a summary of data for each constituent sequence listed below the summary (not shown). This summary includes mapping status and assignment (if applicable), a summary of expression, a list of libraries the sequences were seen in, and a link for each constituent sequence to it's NCBI cluster. The remainder of the cluster report provides a summary of information for each constituent sequence. This includes links to the Sequence Report (described below) which presents a complete set of sequence-related information.

Figure 10 shows a screen capture of the Sequence Report. This interface provides access to all of the sequence feature information that is obtained throughout the sequence processing pipeline. The sequence information is presented in three discrete sections. First, the clone-related information: the clone name and IMAGE ID,[485] if applicable. Next, a summary of the mapping data is presented if applicable. The mapping data includes primers, radiation hybrid retention vector, and


Cluster Report

Home Page Database Search Library/Tissue Search Page Mapping Information Page

Ulowa Cluster: Rn.UI.777
20 Member(s)

Mapping Information	
Number of Placed:	0
Number of Consensus Vector:	0
Number of Primer Ordered:	1
Mapping Source:	MCW

Expression Information

Kidney(x4), NA(x2), Atrium-16.5(x2), ileum(x2), rat-heart-pool(x2), rat-aorta-pool(x2), Bladder(x1), cartilage(x1), Brain(x1), Ventricle-16.5(x1), Eye(x1)

Library Distribution	
TIGR:	5
UI-R-BJ0:	2
UI-R-BJ1:	2
UI-R-CS0:	1
UI-R-CS0s:	1
UI-R-CW0:	2
UI-R-DB0:	1
UI-R-DV1:	1
UI-R-E1:	1
UI-R-EA0:	2
UI-R-Y0:	1

NCBI Cluster Distribution	
NA:	9
Rn.6360:	11
Thiosulfate sulphurtransferase (rhodanese)	

Fig. 9. The cluster report interface.

localization (chromosome, cR position, and placement bin). The web-interfaces are fully cross-linked to the each other.

Submissions of the locally generated EST data are performed on a regular basis to NCBI's dbEST.[87] In addition to the trimmed-for-quality EST sequence, additional information is submitted: (i) presence of polyA tail and polyA signal, including differentiating between canonical and alternative signal; (ii) information pertaining to the tissue from which the EST was derived, as determined by the identified library tag; (iii) how the cDNA library was constructed; and (iv) if any regions of the EST are similar to known repetitive elements. The file format used conforms to the specified "bulk" format described at the NCBI (`http://www.ncbi.nlm.nih.gov/dbEST/how_to_submit.html`) but is not submitted via email due to size constraints. Instead, submissions are made via FTP to a server at the NCBI.

4. Discussion

Because the processing requirements are data parallel, the number of ESTs that can be processed in a single day is easily expanded by adding additional processing nodes. We use PBS/OpenPBS[342] to distribute the processing jobs onto a cluster of workstations. To provide an insight into the relative computational resources required by the various components, a breakdown of the runtime on one thousand sequences is given Figure 11. All times were generated using a single-CPU system with 512MB of available RAM.

To date, we have processed approximately 500,000 EST sequences. This has required significant amounts of both computational and storage resources. Given

Sequence Report

Home Page Database Search Library/Tissue Search Page Mapping Information Page

UI-R-A1-dw-f-02-0-UI.s1 3' EST

Clone Name:	UI-R-A1-dw-f-02-0-UI
Clone IMAGE ID:	1771070
Mapping Information:	
Status:	**Mapped**
	Left primer: GGTCAGGTATGAAGGAAGAGGG Right primer: ACAGCATGACTCCTCCTAAGGG Consensus vector: 2000100021010001000200202200100120210200001001001200001100121000000002000000200010010012120000020011 Chromosome: 4 Map location: 246.36 cR Bin: D4GOT28 - D4RAT21
Map Source:	Ulowa
Ulowa UniGene Cluster:	Rn.UI.779
Tissue Library:	UI-R-A1 (Ulowa), Lib.40 (NCBI)
NCBI UniGene:	Rn.6383
NCBI Entrez:	AA901246 (Accession #)
NCBI dbEST Entry:	4233747 (GI)
Tissue Source:	Lung
Lib Tag:	TTCCA
Poly A Signal:	TTTATT
Poly A Tail Length:	25
Sequence Length:	350

>UI-R-A1-dw-f-02-0-UI.s1 3' EST
TTTTTTTTTTTTTTTTTAAATGGGAAGTATATCAATTTCACTTTATTAGC
CTAATGCAGGAATATAAGGAAAAGGGGATTGAGGTCAGGTATGAAGGAAG
AGGGTTAAAACTTGTGACAGTAAGGATCCTTCAAAACTCGGATGCAGTCC
TCAGTCCTCAGTCACTGCAACGTCACCTCTGCCCACACAAACAGCCACCT
CCGGTTCCTGTCTCCACTTCTGAGCCCAGTAACTCCTCAATCCACGCCCA
GTTACAGTGTGATGAGGTCTACCAGGTTGCCCTTAGGAGGAGTCATGCTG
TCAGGAAAGAAATTCACTAAGGTGTCAAAGAGCTGAGTTCTCTGAGCATA

Fig. 10. The sequence report interface.

the estimates in Figure 11, the complete set of 500,000 sequences would have required 6250 CPU hours to process. In addition, the amount of storage resources required is also prodigious. The chromatograph (SCF) files generated by the ABI software are approximately 190KB each. This represents 95GB of storage. To minimize the storage requirements of the SCF files, we generate a new SCF from the ABI SCF using phred. The benefit of storing the phred SCFs is the significant reduction in average SCF size from 190KB to 40KB. This reduces the amount of storage necessary from 95GB to 20GB.

Although the chromatograph files are the single largest file for any sequence,

Phred	2–3 minutes
ESTprep	2–3 minutes
Masking	1 hour
Annotation	5 minutes
Clustering	5–10 minute
Total	1.25 CPU hours

Fig. 11. Estimated processing time for one sequencing gel.

the remaining files also require a significant amount of storage. These files include several versions of the sequence (original, trimmed, and masked) and quality files (original and trimmed), as well as several files of intermediate annotation (output from RepeatMasker, ESTprep, and BLAST). In all, these files average approximately 100KB to store per sequence.