

Implementation and Testing of an Automated EST Processing and Similarity Analysis System

Elizabeth Shoop^{†§}, Ed Chi[†], John Carlis[†], Paul Bieganski[†], John Riedl[†], Neal Dalton[‡],
Thomas Newman[£], Ernest Retzel[‡]

[†]Computer Science Department, University of Minnesota (§shoop@cs.umn.edu)

[‡]Computational Biology Centers, Medical School, University of Minnesota

[£]DOE Plant Research Laboratory, Michigan State University

Expressed sequence tag (EST) sequencing projects are being undertaken in an effort to identify the function of as many genes as possible from entire genomes. Putative function can be determined by analyzing the similarity of the ESTs to sequences in the public databases. We are involved in a long-term project to research and develop database technology to store and analyze ESTs for Arabidopsis thaliana. The massive amounts of ESTs being produced through automated sequencing technologies necessitates the automated processing and similarity analysis of the ESTs. This paper describes a complete software system that takes ESTs from a sequencing machine, analyzes them for quality, and searches in public databases of previously known sequences. Automating the processing and analysis of the several thousand ESTs produced to date by the Michigan State University Arabidopsis cDNA Sequencing Project has improved the quality of the EST data and the speed at which ESTs can be entered in the public databases. Additional searches that compensate for low complexity regions in the ESTs allowed for more accurate review of the similarity results. The results of the similarity searches are packaged into summarized similarity hit information files with an indication of whether it is possible to identify the ESTs. All processed ESTs and their similarity analysis are available through a Mosaic server, which includes a parsed presentation of the search results, and a three-dimensional graphical display of similarities found. Automating searches in the public databases will make it possible to store putative functional relationships in a database system, in addition to the sequences. An extensible database management system we are developing on a commercial platform will store the data and search results. This will enable biologists to conduct ad hoc exploration using a high-level query language.

1.0 Introduction

We are engaged in a long-term research project to provide a database system to support the collection, analysis, and storage of data from cDNA sequencing projects. The short term objective of this project is the acquisition of EST data. The goal of EST sequencing projects is to sequence enough short sections of expressed cDNA sequence (expressed sequence tags, or ESTs) to obtain the functional expression units of an entire genome, to identify the function of as many genes as possible and discover novel genes. The method used to identify each EST and infer possible function of its corresponding protein is to conduct sequence similarity searches against the public databanks of known DNA and protein sequences [McCo92, Adam92]. The common method used to do this is to run the BLAST similarity search programs [Alts90] for each EST and find "hits" to known sequences, where hits are regions in the EST having a certain degree of similarity to regions in the known sequences. Hits from EST similarity

searches enable inference of probable biological function (often referred to as putative function), whereas a lack of hits implies the possibility of a novel gene discovery.

Motivated by the fact that researchers on EST sequencing projects are producing sequences at such a high rate that manual processing and similarity analysis is virtually impossible, we seek to build a software system that automates each phase of their projects. These phases include processing of the raw sequences, making similarity results available to the rest of the user community, and identifying putative function of each EST. We report here on the implementation and testing of a system that extends previous systems towards the ultimate goal of automated putative function determination. This system consists of these major components:

- A single processing program that is invoked on a set of raw EST sequences. This program sets in motion a series of modules that:
 - 1) check each sequence for quality, trim each sequence to an acceptable level of quality if necessary, and reject low quality ones;
 - 2) trim off the leading vector sequence on each acceptable EST and reject clones that are all vector and have no insert;
 - 3) run *blastx* and *blastn* on each acceptable EST; and
 - 4) translate and check each reading frame for low complexity regions, and run *blastp* when low complexity regions are found.
- A program that displays the alignments for hits in graphical form.
- A program that takes the result of this processing and similarity searching, parses the BLAST output, and creates:
 - 1) a summary file for a set of ESTs, containing all the hits found from *blastn*, *blastx*, and *blastp*, including an indication if the hits are sufficient to possibly identify an EST, and
 - 2) a comprehensive file for each EST, in hypertext markup language (HTML) format, containing all the results of the processing and similarity searches, including hypertext links between the related information and images from the graphical results display program, for viewing on an Internet Mosaic server.
- A program that creates properly formatted files, for a set of acceptable ESTs, for submission to dbEST [Bogu93], which is the public point of submission for ESTs.

In addition to automating each part of EST sequencing projects, we are interested in providing to the community faster and more accurate ways of exploring and finding similarities of interest from the large set of similarity information that is accumulating. To this end, we have designed and begun implementing a database system for similarity results. Using a commercially available database management system with a standard high-level query language, queries in that language will better enable biolo-

gists to find and display similarities of interest.

The principle users of our system are the researchers on the *Arabidopsis* cDNA Sequencing project at the DOE Plant Research Laboratory at Michigan State University. They have produced several thousand ESTs of *Arabidopsis thaliana* to date, and are operating at a throughput of 250 ESTs per week. A second group of users is those in the rest of the biology community who have an interest in *Arabidopsis* EST similarity analysis results. The goal of our system is to meet the needs of the biologists on the *Arabidopsis* EST sequencing project by speeding up the process of EST sequence processing and similarity analysis and to provide results, which can be perused from a graphical interface, to the *Arabidopsis* community in a timely fashion.

The contributions of this work are the following:

- increased automation in filtering high-quality clones and removing leading vector, thereby providing faster throughput of higher-quality ESTs to the community.
- availability of similarity results to the community via an Internet Mosaic server.
- a graphical alignment viewer that enhances the ability to analyze similarity results.
- report of experimental results on our automated method of removing leading vector and detecting poor quality clones for rejection, as well as the effectiveness of LC region masking in aiding identification of ESTs.
- development of an automated method for assigning a value to an EST that provides an indication of whether the similarity hits may allow for its identification, or whether there were not enough hits to do so.
- a description of how a commercially available extensible database management system can be used to allow biologists to more easily find similarities of interest and display them graphically.

In this paper, the description of the processing and analysis system is presented first, then results of experiments performed on 1738 ESTs from *Arabidopsis thaliana*. The design of the similarity result database system is then described. Since this is a system under development, we discuss plans for future improvements.

2.0 Related Work

The BLAST suite of local alignment programs [Alts90] is well-suited to similarity searching for ESTs. In the BLAST suite are the search programs *blastn*, which compares a DNA sequence to a database of known DNA sequences; *blastx*, which compares DNA sequence translated in all reading frames to a database of known amino acid protein sequences; and *blastp*, which compares amino acid sequence to a database of known protein sequences. ESTs represent partially expressed protein sequences that are unlikely to align globally along the length of known protein sequences. The BLAST programs are well-suited to EST similarity searching because they employ a local alignment algorithm that computes scores based on ungapped pairings of elements (nucleic acid bases for DNA or amino acid residues for translated protein) to detect high-scoring local regions of similarity between two sequences. For amino acid sequences, the scores are computed by adding individual scores between two aligned pairs of residues, using a 20x20 substitution matrix with positive and negative values for each possible pair of amino acids, based on the likelihood of an amino acid mutating to another amino acid [Dayh78, Heni93, Alts93]. The same residue pairs aligning are termed "identities," whereas different residues aligning with a positive score from the substitution matrix are termed "positives."

The higher scoring aligned regions of similarity are termed by Altschul et al. as maximal segment pairs (MSPs), and have the favorable property of the ability to compute a statistical significance (p-value from a poisson distribution) for the likelihood of such an alignment randomly occurring by chance within a sequence database of given size and composition [Karl90]. The lower the p-value, the less likely it was that the MSP could have occurred by chance.

Using the BLAST programs, a "hit" can be defined as an unknown EST having a region of similarity to a sequence from the public databanks, because the sequence elements align with a certain strength. The strength is determined by the values of the score and p-value, the percentage of identities, and for amino acid sequences, the percentage of positives. The combinations of these parameters and levels of strength can be varied as more data is gathered about the similarities observed from BLAST for ESTs.

Software systems have recently been developed that attempt to automate portions of the processing and analysis of EST and DNA sequences. Kerlavage, et al. [Kerl93], introduced an EST analysis system that provides tools for allowing users to view and manually edit ESTs, and send them to the BLAST e-mail server for similarity analysis. Tools to aid in manual putative function determination, based on the similarity results, were described, along with a database for storing ESTs and their putative function. Dimidis, et al. [Dimi94] report on a system to automate the detection of vector in DNA sequences and trim the sequences to an acceptable quality level or reject them if they are too short. After this initial processing, their system sends sequences of acceptable quality to the BLAST e-mail server. Results of all processing are written to text document reports after the similarity results are received. Our system is similar to these systems, except that we have further automated steps that were done manually in these systems, and we supply our results in a format that is accessible to the rest of the community over the Internet. Like Kerlavage, et al., our system is designed with the intention of storing the similarity results in a database. Like Dimidis et al., our system automates both sequence quality checking and vector detection, but our system also automates the analysis of BLAST results by computing the strength of each hit and providing an indication of whether the hits are sufficient to possibly identify an EST.

The sometimes-large size of the output files produced by *blastn* and *blastx* can cause difficulty in comparing alignments for various hits, and in comparing results between output files. Biologists need alternate representations of the BLAST output information. In order to provide other representations of hits, the system described in [Kerl93] used a parser of *blastn* and *blastx* output called *btab* [Dubn92], which was used provide views of the hits in a tool for aiding identification of ESTs. The *btab* parser is also used by another system called Blast Output Browser (BoB) [Rao94], which is a graphical user interface for browsing BLAST output. BoB provides access to additional information about the hit sequences, available from the National Center for Biotechnology Information (NCBI) Entrez service [NCBI94]. Our system extends the function of previous BLAST output viewers by parsing results from *blastx*, *blastn*, and *blastp* and re-arranging them into HTML-formatted files for browsing with Mosaic, and by providing the parsed BLAST alignments to an additional tool for graphically displaying and manipulating alignments.

Despite having a high score and low, statistically significant p-value, some hits found in local similarity searches are not biologically significant. These cases frequently occur because the known, naturally occurring biological sequences in the protein databanks have some patterns in them that are not like random

strings of amino acids. Wooten and Federhen [Woot93] note that they have observed that 40% of the entries in the amino acid sequence databases contain local regions of highly biased composition of residues, which are comprised of single tracts of repeated residues, mosaic sequence arrangements with one residue occurring more often, and regular or irregular short-period tandem repeats of some pattern of residues. They developed a method, called SEG, for detecting these regions and report that when these regions are removed from the Swiss-Prot protein databank, the remaining sequences approximate a random distribution of amino acids. These regions of improbably low compositional complexity, often referred to as low complexity regions, will be termed LC regions throughout the rest of this paper. Blastx similarity search results for ESTs containing LC regions can contain hits with alignments of statistically significant scores, considering a random distribution of residues, but of low biological significance, considering how often these regions occur naturally. Moreover, in the case of single residue repeats, multiple hits can often be reported to the same sequence on different reading frames. A review of these issues can be found in [Alts94].

Because these LC regions occur frequently, *blastx* output from ESTs tend to contain many misleading hits (false positives) from which the truly relevant biological hits are difficult to discern. Claverie and States [Clav93] addressed this issue by developing a program called XNU, which detects LC regions in amino acid sequences and masks them out by changing the residues within these regions to an 'X,' which in the substitution matrices used by *blastx* and *blastp* corresponds to an average score from all the possible match or mutation scores in the matrix.

We sought to incorporate the technique of masking out LC regions and discover just how much improvement could be made in the ability to properly analyze the similarity hits and properly identify ESTs. We chose to do this by incorporating XNU into our system and executing additional *blastp* searches on the masked EST sequences. Since we began the work reported here, an option of filtering input sequences through either SEG or XNU has been added to the current version of *blastx* and *blastp*. The results presented below evaluate the usefulness of such filtering.

3.0 Automated Software for EST Processing and Analysis

The necessary processing of EST sequences includes:

- checking quality of clone sequence and trimming to an acceptable level or rejecting clone if unacceptable
- detecting leading vector in clones and trimming it off
- rejecting short clones or clones with sequencing artifacts
- preparing acceptable ESTs for submission to dbEST

Analysis of sequences includes:

- executing a variety of similarity algorithms on ESTs
- checking ESTs for low complexity regions [Clav93, Woot93] and re-executing similarity searches with these regions masked out
- presentation of the results of the analysis in a form that is easily accessible to the user community

The automated sequencing of large quantities of ESTs per day precludes manual processing and analysis. Thus, we have designed a suite of software tools that automate EST processing and analysis and produce results that are easier to peruse and provide insight into the putative function of an EST where possible.

Figure 1 provides an overview of our EST analysis system, showing the software modules, the results they produce, that results from some modules flow into others, and that the final results are gathered into files formatted in hypertext markup lan-

guage (HTML) for use in an Internet Mosaic server. The gray boxes on the left and right in Figure 1 represent single programs that users execute on large sets of EST sequences. These programs automatically produce the following results:

- a summary report on detected similarities for all of the ESTs and whether it may be possible to assign a function to each one, and
- comprehensive individual reports on each EST, containing the results of the processing and similarity analysis.

In the following sections, we outline the function of each module in Figure 1 and point out the specific problems of EST sequence data processing that each of them addresses.

AutoAnalyze: The program on the left in Figure 1, AutoAnalyze, sets in motion a series of modules, whose purpose is described below, including some which execute similarity algorithms on each EST, and sends the user mail when it has finished. The current suite of similarity algorithms executed includes *blastx*, *blastn*, and *blastp*. Gestalt [Bieg94b] is a suffix-tree based alignment algorithm under development in our laboratory, and is to be added to our system in the future. Once AutoAnalyze has completed, the results from the similarity algorithms and other modules are gathered by the SeqSim2html program (on the right in Figure 1) to produce the summary report and the individual HTML-formatted reports.

allncount and nfilter: Each EST needs to be checked for sequence quality and trimmed so that the quality is acceptable or rejected if there are too many unknown base calls. During machine sequencing, the ability to accurately determine successive bases decreases as the length of the sequence increases. Problems occurring during laboratory preps sometimes cause inability of the sequencing machine to accurately determine bases. A simple measure of quality of a sequence is the ratio of unknown bases to the total number of bases in that sequence. A sequence with less than 5% of unknown bases is considered to be of reasonably good quality.

Allncount produces a rough indication of whether a batch of ESTs is of acceptable quality by producing a histogram of percentage of sequences with unknown bases (indicated by an N) at each position.

Nfilter checks the percentage of N bases in an individual clone/EST and trims them to an acceptable quality level if necessary. If the clone is trimmed down to 300 base pairs and its quality remains unsatisfactory it is rejected. We chose 300 bp because previous work has shown that the ability of BLAST programs to find all similar database sequences is diminished for sequences less than 300 bp long [Shoo94].

gstVF: Clone sequences from the automated sequencer contain leading vector sequences. If the vector sequence is not trimmed off, the true potential homologies in similarity search results are overshadowed by matches to vector sequences in the public databanks [Gish93].

gstVF uses generalized suffix trees [Bieg94a] to quickly locate the vector sequence within a raw clone and trims off the leading vector to produce the true EST. Clones in which the cDNA fragment is very short or did not get properly inserted in the vector are detected and discarded. This program eliminates the burden of manual editing of ESTs, and avoids the problem of spurious similarity search hits.

std2seq: Most similarity search programs require files containing sequences to be in 'FASTA' format [Pear88]. Std2seq takes the raw EST sequence files from the sequencer and translates them into 'FASTA' formatted files.

LCFilter (mreppfilter + mrepprocess): ESTs might con-

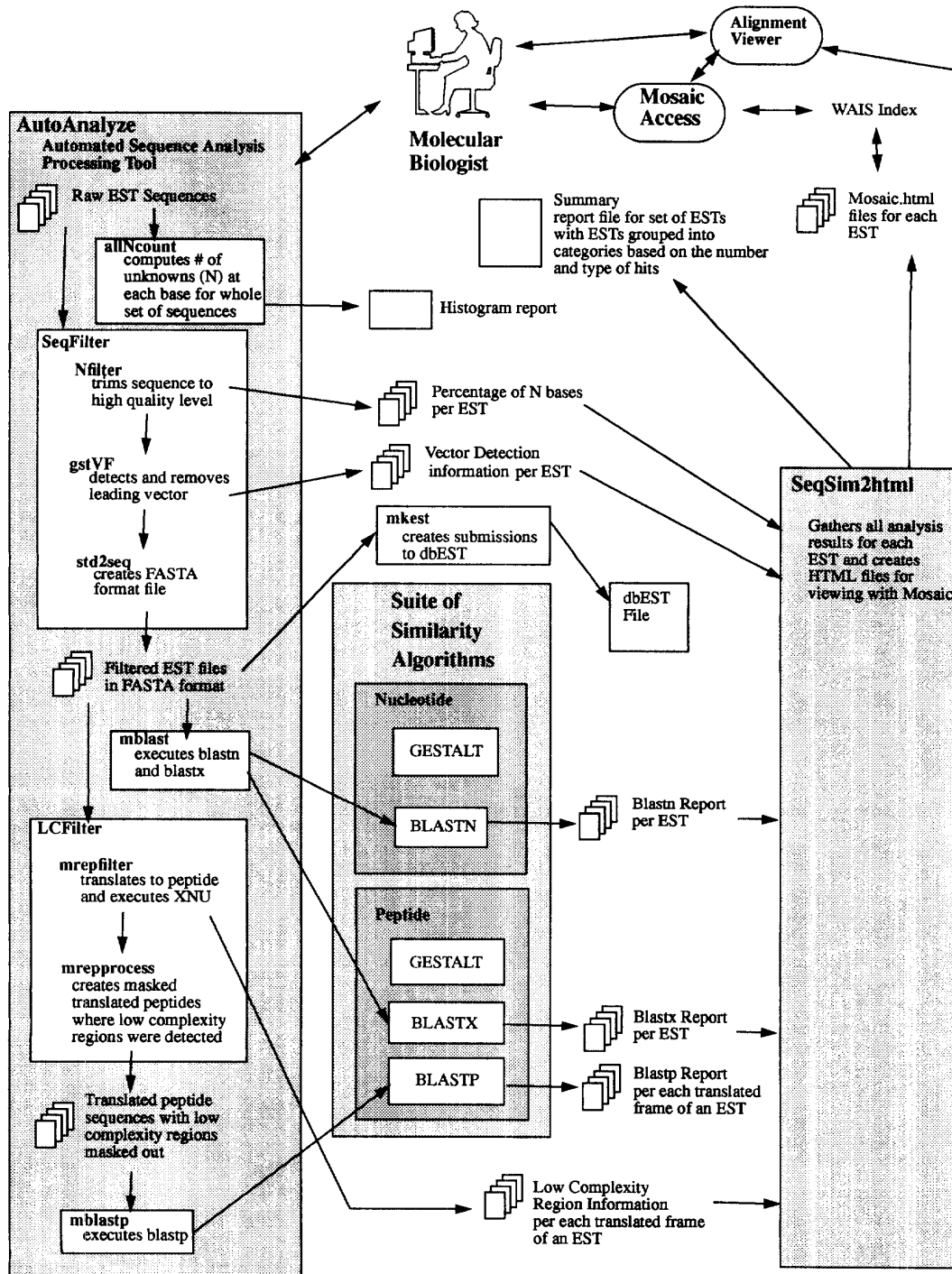


Figure 1. System Overview

tain regions of low complexity, which can potentially show up as misleading high-scoring regions in similarity algorithm results (false positives). The two modules making up the low complexity filter (LCFilter) use the XNU program of Claverie and States [Clav93] to identify potential low complexity regions, and mask out those regions in peptide sequences translated from the ESTs.

mblast, mblastp: Since there are many ESTs being processed at a time, it is too time consuming to manually run similarity programs such as *blastx* and *blastn* on each EST individually. The *mblast* script repeatedly executes *blastx* and *blastn* for each EST file given on the command line, and *mblastp* runs *blastp* for any translated frame of the original EST given on the command line that had low complexity regions masked out. Both *mblast* and *mblastp* monitor the load average on the machine so as to utilize the machine completely but not unduly disturb other users.

SeqSim2html: The results of the automated analysis performed on a set of ESTs consists of several files that contain information about each individual EST. Some of the files, like the similarity search outputs, are hundreds of lines long and are inter-related. It is difficult for users to manually peruse each file for an EST and get the whole picture of the analysis for that EST. It is equally difficult to obtain a summarized view of the results of the similarity searches for a set of ESTs.

For a given set of ESTs, the SeqSim2html program packages the results from the various analyses performed and produces a single document for each EST with hypertext links connecting related material. Users can browse this single document using Mosaic and move through it using the links. For each hit, links to the NCBI World Wide Web (WWW) Entrez server are made, using the accession number of the public databank sequence. SeqSim2html also produces a consolidated report for the entire set of ESTs, listing all relevant hits and their relative strength, from the *blastp*, *blastx*, and *blastn* results. Since this analysis process can take over an hour for a large set of ESTs, SeqSim2html informs users by sending them e-mail when it has completed.

SeqSim2html also produces a listing of the category that each EST falls into, based on the hits produced by *blastx* and *blastp*. SeqSim2html parses the *blastn*, *blastx*, and *blastp* output, and filters out irrelevant hits, based on input parameters prescribed by the users. Since ESTs are usually directionally cloned, users can choose to have only plus or minus strand hits displayed (or both if the direction is unknown). Users can set a cutoff for the worst p-value accepted as a legitimate hit. These choices are offered so that during post-processing, different levels of acceptance from what were chosen during original BLAST runs can be experimented with. The *blastx* and *blastp* hits are also assigned a level of strength¹, according to the following criteria:

- *strong* hits have score > 150 and p-value < 0.005
- *nominal* hits have score > 80 and p-value < 0.01
- *weak* hits have score <= 80 and p-value < 0.01

These values were chosen after reviewing the *blastx* search results from approximately 100 ESTs and determining that the hits for these ESTs fell naturally into these levels, which were useful in determining whether the hits provided enough informa-

1. The values chosen for score are dependent upon the substitution matrix used. For this study we used the PAM250 matrix. The values used for these categories can be varied. In the future, the use of score may be eliminated entirely. We found it necessary for the cases where the *blastx* algorithm attempted to compute a combined P-value for several MSPs found in the same sequence, where the score was relatively low for each MSP and the combined P-value was inflated.

tion to possibly identify the EST by inferring its function. Using these computed levels of strength for the hits associated with each EST, we then were able to automatically compute a category into which each EST should fall, which describes whether or not it should be possible to infer its function. The categories computed for each EST are as follows:

Categories 1 and 2 are comprised of ESTs in which no LC regions were found within their *blastx* hits, and:

- Category 1: they had hits enough to either determine function or to warrant further investigation, or
- Category 2: they had few or no good hits.

The remaining two categories are comprised of ESTs where LC regions were within hits that *blastx* found:

- Category 3: using the *blastp* results, filtering out any misleading hits aided in determining that this EST fell into Category 1 or 2 above, or
- Category 4: *blastp* results were not different from the *blastx* results

For the above categorization, we chose the following criteria, again based on our observation of approximately 100 ESTs, this time using the summarized outputs that SeqSim2html produces, containing the strength of each hit. For an EST to fall into category 1, it had to have:

- at least one strong hit with greater than 40% identities, or
- 2 or more nominal hits with greater than 30% identities

Any other ESTs without LC regions would fall into category 2. Category 3 ESTs were those where there were LC regions within *blastx* hits, and the *blastp* results for those ESTs showed fewer hits. Using the *blastp* results and the above criteria, the EST could subsequently fall into category 1 or 2. Depending on whether the EST could now be placed in category 1 or 2, it was designated as category 3a or 3b, respectively. Those ESTs where *blastp* results were not different from *blastx* were placed in category 4.

These categories have proven to be quite useful, as we can now separate the ESTs into two major groups: those whose function we ought to be able to determine, and those who need further examination and subsequent periodic similarity searching as the public databanks are updated.

AlignmentViewer (av): The textual output from similarity search algorithms such as *blastp*, *blastx*, and *blastn* can be difficult to interpret when there are several hits to different regions of an unknown input sequence. The hits have varying alignments, scores, and p-values and occur in different frames. When the hits are ordered by score or p-value, the ones in the same frame and to the same region of the unknown are sometimes scattered throughout the list, making detection of the best true hits difficult. The AlignmentViewer(av) tool provides a 3-dimensional graphical representation of all the hits for an unknown input sequence, separating the hits by frame and organizing them by score and region along the unknown sequence. The resulting display provides for quicker interpretation of the *blastx* or *blastp* hits most likely to provide clues about the function of the unknown sequence. Figure 2 depicts examples of the features and use of the AlignmentViewer tool. In Figure 2a, the longer Y-axis corresponds to the scores of the alignments for the hits, the X-axis represents the residues along the length of the EST, and the Z-axis is for the relative strength of each individual residue pairing, taken from the substitution matrix used by *blastx* or *blastp*. In Figure 2a, a series of hits of varying strength can be seen, with the negative-scoring bases projecting out of the X-Y plane in white, and the positive-scoring bases projecting in and perpendicular to the X-Y plane in black.

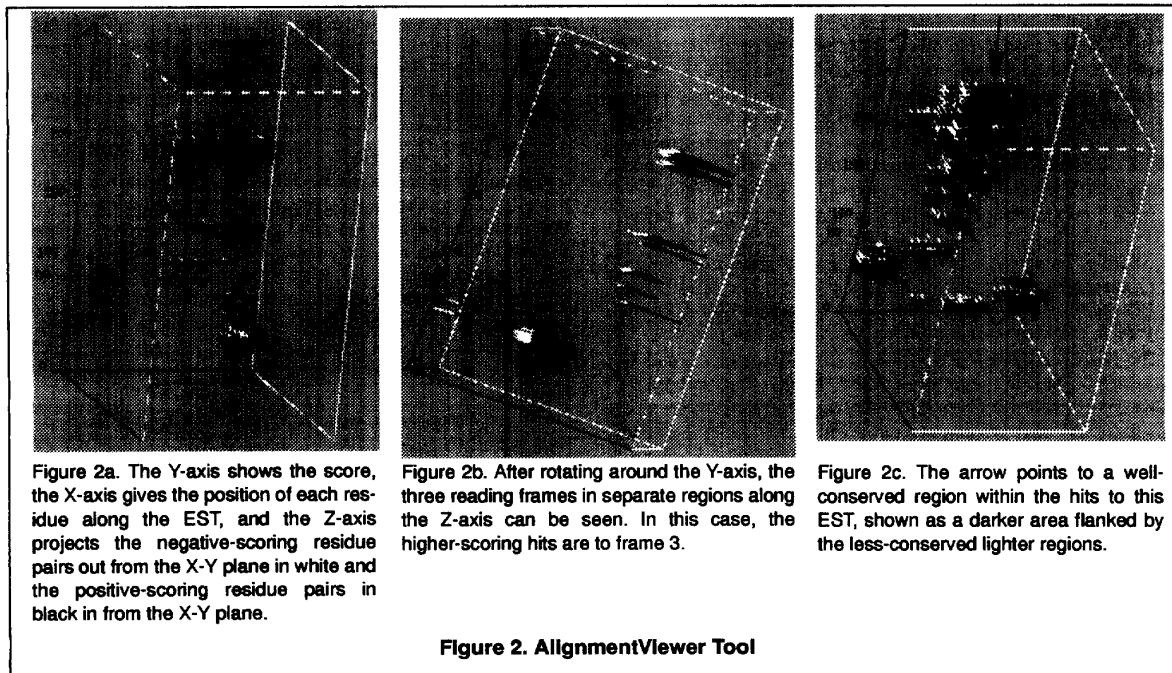


Figure 2b shows how the hits on various frames are shown at different depths along the Z-axis, thus revealing that the best-scoring hits were from frame 3. The AlignmentViewer tool is implemented in color, with different colors for positive and negative hits and different pairs of colors for each frame. This provides good contrast and visual cues for the user. The AlignmentViewer tool is also useful for recognizing conserved regions in a set of hits. Another example from a different EST is given in Figure 2c to depict this. The arrow points to the conserved region in these hits, which appears darker than the less conserved regions flanking it.

The AlignmentViewer tool thus allows biologists to get a better indication of the total number of hits from the textual output, their strength, where they lie within the EST, and where the well-conserved regions are located. This tool also proved useful for comparing *blastx* runs to *blastp* runs where LC regions have been masked out, which will be discussed in Section 4.0, where some experimental results are presented.

Mosaic Access to EST Analysis: The biologists interested in the EST sequence analysis performed by our system are located throughout the world, yet they require timely access to this information. We collaborate closely with researchers in the sequencing lab for the *Arabidopsis* cDNA Sequencing Project at Michigan State University, who require daily access to the data. Other researchers interested in *Arabidopsis* EST similarity analysis also need to access and search this information resource for similarity results of interest. All of the sequence analysis results, including images from AlignmentViewer, are available on an Internet Mosaic server, thus providing access to the research community. Users can navigate within an EST document using the hypertext links that have been built into the files.

In an effort to provide easier access to EST similarity results of interest to our users, we have also built a Wide Area Information Servers (WAIS) index [Kahl92, NISO88] on the HTML-for-

matted files, allowing simple keyword search queries that return EST result files containing those keywords.

mkest: The large numbers of ESTs produced necessitates automating the creation of the specially-formatted files to be submitted to dbEST[Bogu93]. The special format required also means that producing these files manually is prone to error.

For a given set of EST files and pre-defined information about the lab producing the ESTs, the mkest program automatically generates the properly formatted files for submission to the dbEST database. This program has been used successfully to submit over 5000 *Arabidopsis* ESTs.

4.0 Experimental Results

In order to assess the value of this automated system and evaluate the results of the EST similarity searches, we tested it on two groups of ESTs: 1). 333 raw ESTs from the sequencing machine, known to have varying levels of quality, and 2). 1519 EST sequences which had been edited manually to remove the leading vector sequence. These sequences were originally processed prior to the completion of the system presented here. Because of the manual editing, the quality level on a few of these ESTs does not meet what we now consider acceptable.

For the first group, we wished to assess the performance and value of AutoAnalyze in general, and allNcount and SeqFilter in particular. AutoAnalyze was executed on the original raw sequences as they came from the sequencing machine. It completed each individual execution of the modules within it without fail, and sent an e-mail message when it was completed. AllNcount produced a textual histogram output that gives quick indication of where the majority of the unknown bases are occurring in the sequences. Within the SeqFilter module, Nfilter rejected 114 of the ESTs as having greater than 5% unknown bases, even after being trimmed to 300 bp. Of all of the remaining 219 ESTs, gsvf was able to detect and trim the leading vector sequence.

For the two groups of sequences together, we wished to assess the benefit and performance of SeqSim2html, and examine the results of the categorization of the ESTs, based on their similarity search hits. SeqSim2html ran without fail on all 1738 EST sequences, producing a summary report, individual comprehensive reports, and a list placing each EST into a category (as described in Section 3.7). The categorization was checked manually for 50 of the sequences, to ensure that the computed method was correct. The results of this categorization appear in Table 1.

Table 1 divides the processed ESTs into four categories. Many ESTs did not have LC regions within their *blastx* hits, and are classified as category 1 or 2. ESTs that did have LC regions within their *blastx* hits are classified as category 3 or 4. Masking the LC regions improved the indication of function for ESTs in category 3, whereas masking did not for those in category 4.

The results in Table 1 show that 21.67% (categories 3 and 4) of the ESTs had LC regions detected by the XNU program¹ within the hits found by *blastx*, and that the additional *blastp* runs with these LC regions masked out allowed for 17.95% of the ESTs to have misleading hits filtered out (category 3). Thus, proper identification of meaningful hits was possible, and the addition of the LCFilter and mblastp modules into our system allowed for more accurate review of *blastx* similarity results.

The ESTs that fell into category 3 in Table 1 are divided into A) those in which enough positive hits remained in the *blastp* results to reveal a possible function, and B) those where no hits remained. We cannot rule out the possibility that the masking of LC regions by XNU caused possible true positive hits to be eliminated. For this reason, we chose to keep both the *blastx* results and the *blastp* results and present them to users, along with information about whether an LC region occurred in the *blastx* hits.

A statistical analysis of the results in Table 1 reveals the confidence with which we expect the ratio of ESTs that fell into category 3 versus those that fell into category 4. Recall that category 3 sequences are cases where XNU and *blastp* successfully aided our analysis by more accurately determining if an EST could be identified or not. On the other hand, XNU and *blastp* failed to aid our analysis in category 4 sequences. Using the experimental results shown in Table 1, the 375 ESTs that fell into category 3 and 4 can be considered approximately as a binomial random distribution of what we would expect to see in the entire population of *Arabidopsis* ESTs with LC regions. With category 3 marked as success, the probability of success in this binomial experiment is $p = .832$ (312 in category 3 out of 375 total). We can then determine our confidence in this probability. Since the total number of sequences in category 3 and 4 is $n = 375$, and $n * p > 5$ and $n * (1 - p) > 5$, this binomial experiment can be approximated with a normal distribution [Devo87]. Using standard 95% confidence interval formula, the constructed 95% confidence interval is (.812, .851). Therefore, we can be 95% confident that the number of ESTs having LC regions and falling in category 3 will be between 81% and 85%.

An illustration of the power and utility of both the Alignment-Viewer tool and the use of XNU to mask out LC regions can be shown in Figure 3. For one EST, Figure 3a shows the many hits from the *blastx* results, in both the second and the third frame, and

1. After review of original test executions of *blastp* on the masked sequences produced by XNU, we decided that using the default probability *p*-value of 0.01 for the XNU program was not discriminating enough and detected too many LC regions that did not contribute to misleading *blastx* hits. Lowering this value to a more discriminating 0.005 *p*-value produced the results shown in Table 1.

below it in Figure 3c are the summary of the *blastx* results from the Mosaic EST file. Figure 3b shows the AlignmentViewer depiction of the *blastp* results after XNU masked out the LC regions of that same EST. The true hits are revealed, and those hits are shown, in Figure 3d, in the summary of the *blastp* results from the Mosaic file.

5.0 Database Design for Similarity Results Analysis

Having developed a system for automatically generating summary and comprehensive reports from EST processing and similarity analysis, we now see that biologists will become inundated with these reports and will require methods to explore and query this large amount of similarity information for similarities of interest. We therefore have designed a database schema for similarity results, to be implemented on Illustra [Ston93, Ubel94], an extensible object-relational database management system (DBMS), based on the POSTGRES research prototype DBMS [Ston90].

Figure 4 depicts the schema for the sequence similarity database, using the logical data structure (LDS) notation [Held89]. The LDS in Figure 4 depicts the nature of the similarity result data, independent of how it is to be stored. We use this DBMS-independent LDS notation because an LDS diagram can be mapped to many different data representations, including, but not limited to, those of relational DBMSs. In the LDS in Figure 4, data entities are in boxes with their names at the top, and attributes of the entities are inside the boxes, with the unique identifying attribute of the entity underlined. Relationships between the entities are given by lines between the boxes. The "chickenfoot" symbol at the end of some relationship lines represents the 'many' end of one-to-many relationships. Lines without chickenfeet represent one-to-one relationships.

A brief overview of the LDS schema in Figure 4 follows. The DBMS will contain sequences and related laboratory information to aid researchers who have large amounts of sequence data to be analyzed for similarity. The type of information the EST sequencing project researchers may want to keep about each EST, for example, is EST identifier, nucleic acid sequence, name of clone, vector used, primer used, date sequenced, sequence quality, and low complexity regions of the sequence.

The DBMS will store the results from the similarity searches, including, but not limited to the algorithms that have been executed, the input parameters used, the databank sequences which had some similarity ('hits') and which databank they came from, the alignment produced, the score, and the statistical significance of the score (*p*-value).

Implementation of the database schema shown in Figure 4 is in progress, using the Illustra DBMS. We chose Illustra principally because it allows addition of user-specified operations on data that can be placed in queries using a standard DBMS-supplied SQL query language. We will use this feature to add special operations, such as testing for the significance of a particular hit, or graphically displaying hits, or specialized keyword searches on the free-form text descriptions of public databank sequences.

Using the entity and attribute names shown in Figure 4, some SQL queries that can be constructed with these operations will now be presented. The following query would retrieve similarity results for all the sequences in the database which have them, where the score was at least marginally significant.

```
SELECT  AlgorithmName, ParamSet#, SequenceId, PubDbSeqDe-
        scription, PubDbSeqId
FROM    SimilarityResult, Algorithm, ParameterSet, Hit, Public-
        DatabankSequence
WHERE   (Significant(P-value) OR MarginallySignificant(P-value))
ORDERBY PubDbSeqId
```

Table 1: Results of Automated Categorization of ESTs

ESTs With No LC Regions Within Blastx Hits		ESTs with LC Regions Within Blastx Hits (Blastp Searches Were Executed)		
Category 1	Category 2	Category 3		Category 4
Enough hits to have possible indication of function	Not enough hits to determine function	A Blastp hits removed some misleading <i>blastx</i> hits so that there was a possible indication of function	B Blastp hits indicated there were not enough true hits to determine function	Blastp hits were the same as <i>blastx</i> hits
648	715	80	232	63
37.28%	41.14%	4.60%	13.35%	3.62%

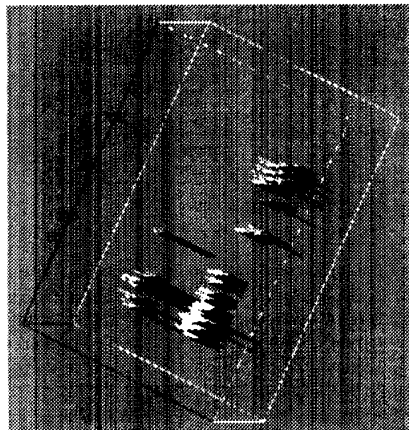


Figure 3a. The original *blastx* hits for an EST, with some in frame2 and some in frame 3. Some of these are misleading false-positives.

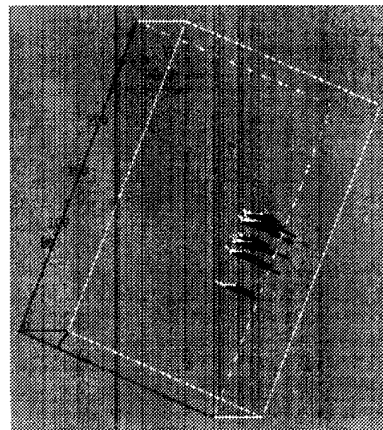


Figure 3b. The *blastp* hits after XNU had masked out the LC regions. The true-positives are revealed.

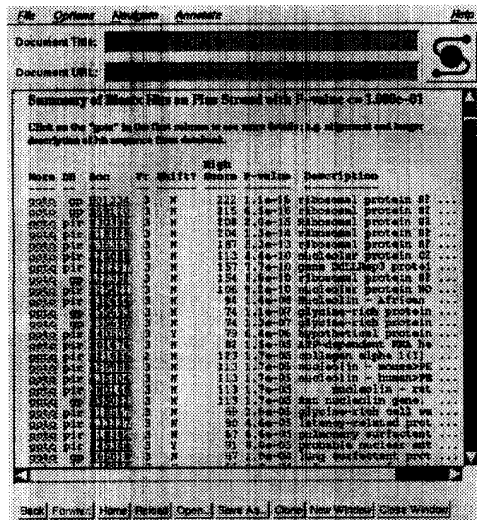


Figure 3c. The Mosaic interface to this EST, with the summary of the *blastx* hits shown. Note the hits to "glycine rich" proteins.

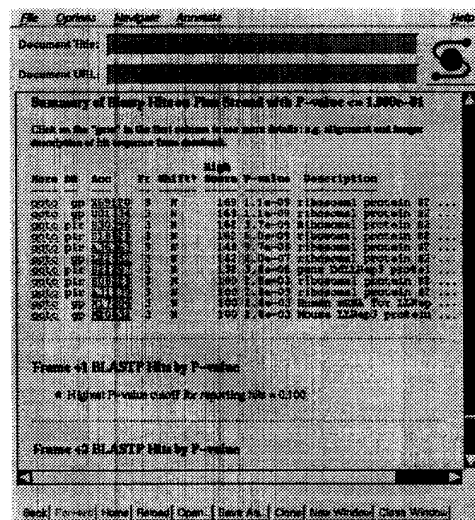


Figure 3d. The Mosaic interface to this EST, showing the summary of *blastp* hits after LC regions masked out. Note the many misleading hits are removed.

Figure 3. Differences between Original *blastx* output and *blastp* after LC regions masked out.

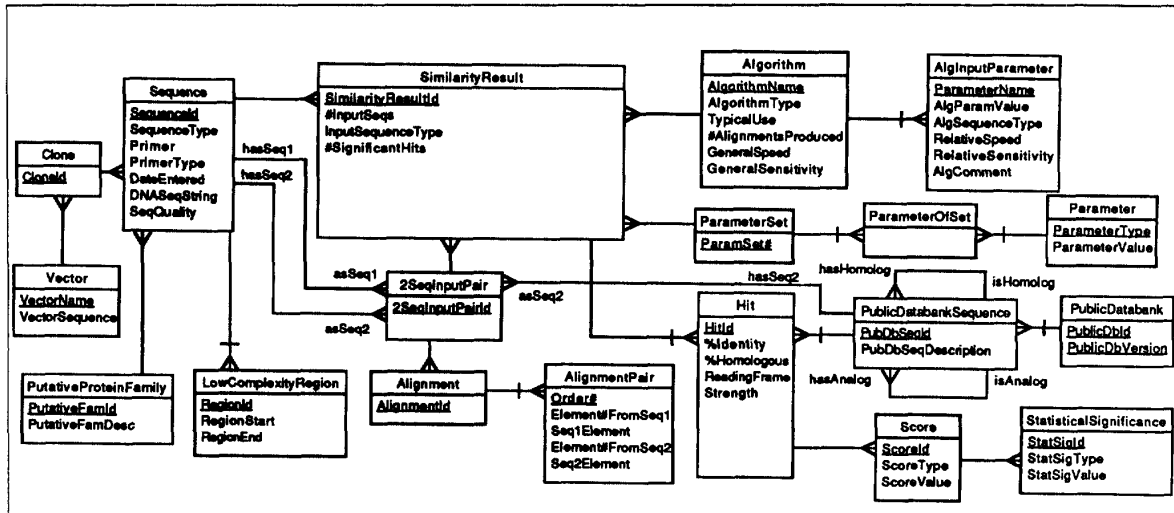


Figure 4. LDS Schema of the Similarity Results Database.

The functions *Significant* and *MarginallySignificant* will be provided so that users will have some common reference for what a significant P-value should be. The ORDERBY feature of SQL is used here so that the user can get an indication of which algorithms found similarities to a particular sequence in the databank and which did not. Now if a single alignment for one sequence against a 'hit' sequence in the databank is to be displayed, the query would be:

```
SELECT AlignmentViewer(AlignmentId)
FROM Sequence, SimilarityResult, 2SeqInputPair, PublicData-
bankSequence, Alignment
WHERE InputSequenceId = 37C5T7 AND HitSeqId = HML4406
```

The *AlignmentViewer* function is the previously described GUI application that would get displayed and could be manipulated by the user.

A user who was interested in particular types of proteins will want to know which sequences 'hit' those types of proteins in the public databanks. One possible method for this type of exploration is by allowing users to search for keywords in the descriptions provided for sequences in the public databanks. An example query using this capability is:

```
SELECT SequenceId, PubDbSeqDescription, PubDbSeqId
FROM Sequence, SimilarityResult, Hit, PublicDatabankSequence
WHERE (KeywordSearch(PubDbSeqDescription, "kinase or cytok-
ine"))
ORDERBY PubDbSeqId
```

These types of queries could be executed again and again for different sequences and alignments and various display applications could be running simultaneously, allowing the user to view results from many similarity searches at once.

When completed, this database for similarity results will enable biologists to more effectively narrow the large data space of information to similarities of interest and explore those similarities with the aid of graphical displays.

6.0 Future Directions and Enhancements

The system we have described is functional, and in use on a daily basis. However, it is evolving. The areas that we are still addressing include refining the quality of both the raw and the distributed data, enhancements in putative function assignment, additions to similarity search tools, and the development of an

underlying database manager. The quality of the raw data could be improved by addressing not just the number of ambiguous base calls, but by developing statistical measures of sequence quality. In addition, highly abundant cDNA's, while dramatically reduced by the techniques utilized in the preparation of the libraries, remain somewhat over-represented in the anonymous clones developed to sequences. This over-representation is reflected in the data distributed to dbEST, and could be minimized by developing a generalized suffix tree of raw data for the detection of such sequences. Given that the ESTs can now be categorized as to whether they can possibly be identified or not, each of these two groups can be further analyzed separately. The ESTs that can be identified will be analyzed with the aid of the *AlignmentViewer* tool and other tools to guide quick putative function assignment. For those ESTs that have insufficient hits, methods for automatically re-executing similarity searches periodically as the public databanks get updated will be developed, as well as alternate methods of identification, including gene detection [Fiel90, Fick92, Guig92, Stad90, Uber91] and more sensitive similarity algorithms. As an alternate means of analysis, the Gestalt algorithm will be used as a real-time similarity query method in the database system.

7.0 Conclusion

The system presented here is in use by researchers in the *Arabidopsis* cDNA sequencing project at Michigan State University, and has proven to be a valuable improvement over the previously manual and mind-numbing task of editing individual EST sequences and executing BLAST searches on each, along with the daunting and error-prone task of poring over multiple similarity report files in order to glean information on each one.

The system described here enhances the capabilities of other systems [Ker193, Dimi94] with the following features:

- an automated method for the removal of leading and trailing vector sequences;
- local execution of similarity searches gives us control of the databases, algorithms, parameters and matrices used; local execution also avoids inundating the NCBI server with large numbers of sequences, and provides faster turnaround of results;

- the system has a method for automatically detecting LC regions, as well as for re-executing similarity searches with LC regions masked;
- an automated method for assigning a value to an EST that provides an indication of whether the similarity hits may allow for its identification, or whether there were not enough hits to do so;
- a similarity search results parser that produces hypertext-linked Mosaic files for Internet data access to the rest of the community;
- a graphical alignment viewer, in addition to the textual output from the various algorithms.
- an automated method for creating files properly formatted for submittal to the dbEST database.

This system is geared towards as much automation as possible, while maintaining, organizing and flagging all results for inspection by researchers. This increased automation reduces the number of error-prone steps (e.g., sequence editing and quality assessment). By reducing errors in the data processing and manipulation, we increase the quality of data submitted to the public databases. The completed design and implementation of the database for similarity results, similarity methods, tools for hypertext and graphical representation of this information, and high-level query language for ad-hoc querying will enable biologists to search for sequences of interest in ways that have been difficult, if not impossible, previously.

8.0 Data Analysis and Software Availability

The completed similarity results from *Arabidopsis* ESTs are available on the following Internet Mosaic server:

<http://lenti.med.umn.edu>

The software for the system, written in Perl, C, and C++ for Sun workstations and a Sun SparcServer is also available on ftp site lenti.med.umn.edu.

9.0 References

- [Adam92] M. A. Adams, et al. "Sequence identification of 2375 human brain genes." *Nature*, 355:632-634, 1992.
- [Alts90] Stephen Altschul, Warren Gish, Webb Miller, Eugene Myers, and David Lipman. "Basic local alignment search tool." *Journal of Molecular Biology*, 215:403-410, 1990.
- [Alts93] Stephen Altschul. "A protein alignment scoring system sensitive at all evolutionary distances." *Journal of Molecular Evolution*, 36:290-300, 1993.
- [Alts94] Stephen F. Altschul, Mark S. Boguski, Warren Gish, and John C. Wooten. "Issues in searching molecular sequence databases." *Nature Genetics*, 6:119-129, 1994.
- [Bieg94a] Paul Bieganski, John Riedl, John Carlis, and Ernest Retzl. "Generalized suffix trees for biological sequence data: Applications and implementation." In *Proceedings of the 27th Annual Hawaii International Conference on System Sciences*, volume 5, pages 35-44. IEEE, IEEE Computer Society Press, 1994.
- [Bieg94b] Paul Bieganski, John Riedl, John Carlis, and Ernest Retzl. "Gestalt: Content-directed sequence database searching using generalized suffix trees." Manuscript in progress, 1994.
- [Bogu93] M.S. Boguski, T.M.J. Lowe, and C.M. Tolstoshev. "dbest - database for expressed sequence tags." *Nature Genetics*, 4:332-333, 1993.
- [Clav93] Jean-Michel Claverie and David States. "Information enhancement methods for large scale sequence analysis." *Computers and Chemistry*, 17(2):191-201, 1993.
- [Dayh78] M. O. Dayhoff, R. M. Schwartz, and B. C. Orcutt. "A model of evolutionary change in proteins." In M. O. Dayhoff, editor, *Atlas of Protein Sequence and Structure*, Vol. 5, Suppl. 3, chapter 22, pages 345-352. National Biomedical Research Foundation, 1978.
- [Devo87] Jay L. Devore. *Probability and statistics for engineering and the sciences*. Brooks/Cole Publishing, 2nd edition, 1987.
- [Dimi94] Stamatis Dimidis, Nabil Kamel, and John Dame. "Design and implementation of a DNA sequence processor." In *Proceedings of the 27th Annual Hawaii International Conference on System Sciences*, volume 5, pages 98-107. IEEE, IEEE Computer Society Press, 1994.
- [Dubn92] M. Dubnick. "Btab - a BLAST output parser." *Computer Application in the Biosciences (CABIOS)*, 8(6):601-602, 1992.
- [Gish93] Warren Gish and David States. "Identification of protein coding regions by database similarity search." *Nature Genetics*, 3:266-272, 1993.
- [Held89] J. Held and J. Carlis. "Conceptual data modeling of an expert system." *IEEE Expert*, Spring 1989.
- [Hen93] Steven Henikoff and Jorga Henikoff. "Performance evaluation of amino acid substitution matrices." *Proteins: Structure, Function, and Genetics*, 17:49-61, 1993.
- [Kahl92] Brewster Kahle, Harry Morris, Franklin Davis, Kevin Teine, Clare Hart, and Robin Palmer. "An executive information system for unstructured files: Wide area information servers." *Electronic Networking*, pages 59-68, 1992. Available via anonymous ftp at ftp.wais.com: /pub/wais-inc-doc.
- [Karl90] Samuel Karlin and Stephen F. Altschul. "Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes." *Proceedings of the National Academy of Sciences*, 87:2264-2268, 1990.
- [Kerl93] Anthony R. Kerlavage, Mark D. Adams, John C. Kelly, Mark Dubnick, John Powell, Pari Shanmugam, J. Craig Venter, and Chris Fields. "Analysis and management of data from high-throughput expressed sequence tag projects." In *Proceedings of the 26th Annual Hawaii International Conference on System Sciences*, volume 1, pages 585-594. IEEE, IEEE Computer Society Press, 1993.
- [McCo92] W. Richard McCombie, et al. "Caenorhabditis elegans expressed sequence tags identify gene families and potential disease gene homologues." *Nature Genetics*, 1:124-131, 1992.
- [NCBI94] National Center for Biotechnology Information (NCBI). *Entrez Application Version 1.9*. entrez@ncbi.nlm.nih.gov. <http://www.ncbi.nlm.nih.gov/Search/Entrez/index.html>.
- [NISO88] National Information Standards Organization (NISO). *American National Standard Z39.50, Information Retrieval Service Definition and Protocol Specifications for Library Applications*. Transaction Publishers, New Brunswick, NJ, 1988.
- [Pear88] William R. Pearson and David J. Lipman. "Improved tools for biological sequence comparison." *Proceedings of the National Academy of Sciences*, 85:2444-2448, 1988.
- [Rao94] Parasa Venkateswara Rao and John Powell. "Bob (BLAST output browser), release 1.01." Available via ftp on milo.dcrn.nih.gov, 1994.
- [Shoo94] Elizabeth Shoop, John Carlis, and Ernest Retzl. "Automating and streamlining inference of function of plant ests within a data analysis system." In *Proceedings of the 27th Annual Hawaii International Conference on System Sciences*, volume 5, pages 47-48. IEEE, IEEE Computer Society Press, 1994.
- [Stad90] Roger Staden. "Finding protein coding regions in genomic sequences." *Methods in Enzymology*, 183:163-180, 1990.
- [Ston90] Michael Stonebraker, Lawrence Rowe, and Michael Hirohama. "The implementation of postgres." *IEEE Transactions on Knowledge and Data Engineering*, 2(1):125-142, 1990.
- [Ston93] Michael Stonebraker. "The miro database." *SIGMOD Record - Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, 22(2):439, 1993.
- [Ubel94] Michael Ubell. "The Montage extensible database architecture." In *Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data*, page 482, 1994.
- [Woot93] John C. Wooten and Scott Federhen. "Statistics of local complexity in amino acid sequences and sequence databases." *Computers and Chemistry*, 17(2):149-163, 1993.