

**The Design and Implementation of an EST Annotation System
for a Fungal Genomics Project**

Jian Sun

A Thesis
in
The Department
of
Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Computer Science at
Concordia University
Montreal, Quebec, Canada

March 2005

© Jian Sun, 2005



Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file *Votre référence*

ISBN: 0-494-04451-9

Our file *Notre référence*

ISBN: 0-494-04451-9

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.


Canada

Abstract

The Design and Implementation of an EST Annotation System for a Fungal Genomics Project

Jian Sun

Expressed Sequence Tags (ESTs) are tiny pieces of DNA sequences that are generated by sequencing from either the 5' or 3' end of an expressed gene. They provide a highly cost-effective method of accessing and identifying expressed genes. The large-scale EST sequence data, generated in the Concordia fungal genomics project, raise the following two critical questions that need to be answered by their bioinformaticians. First, from the biologists' point of view, what is the most useful information encoded in these sequences, and how can that information be obtained? Second, what is the most efficient way for the scientists to access this kind of information? In this thesis, we provide our answer to the above questions by building an EST annotation system that can fulfill the tasks for EST data storage, function assignment and visualization.

Annotation is the processes of adding biological information to EST sequences. The contents of an EST sequence annotation system could vary from one system to another; however, based on an actual genomics research project, our EST annotation system not only presents most of the common features in genomics annotation systems, but also includes a number of specific functions that are very useful for EST projects, such as sequence full-length prediction, sequence-based and unigene-based data integration, and protein signal peptide prediction. The methods used and described in this thesis could be a very useful protocol for most EST sequence annotation systems.

TABLE OF CONTENTS

1. INTRODUCTION	1
1.1 FUNGAL GENOMICS PROJECT OVERVIEW.....	1
1.2 THE STATEMENT OF PROBLEMS.....	2
1.3 MAIN ACHIEVEMENT OF OUR CURRENT WORKS	3
1.4 CONTRIBUTION OF THIS THESIS	4
1.5 LAYOUT OF THE THESIS	5
2. BACKGROUND	6
2.1 BIOLOGICAL CONCEPTIONS AND TECHNIQUES.....	6
2.1.1 <i>DNA, mRNA and cDNA</i>	<i>6</i>
2.1.2 <i>cDNA library construction, Cloning.....</i>	<i>9</i>
2.1.3 <i>EST method and Sequencing.....</i>	<i>9</i>
2.2 EXISTING EST ANALYSIS PIPELINES.....	13
2.2.1 <i>Main steps of EST data analysis process.....</i>	<i>13</i>
2.2.2 <i>Existing EST analysis pipelines</i>	<i>15</i>
2.3 SEQUENCE ANNOTATION SYSTEM OVERVIEW	20
2.3.1 <i>Main goals of sequence annotation system.....</i>	<i>20</i>
2.3.2 <i>Existing sequence annotation systems</i>	<i>21</i>
2.3.3 <i>Basic requirement for presenting our EST data.....</i>	<i>34</i>
3. METHODS USED	36
3.1 ANNOTATOR: A BASIC SEQUENCE ANNOTATION PROGRAM	36
3.2 SEQUENCE FRAMES AND TRANSLATION.....	39
3.3 NCBI BLAST FOR SEQUENCE SIMILARITY	40
3.4 MOTIF SEARCH AND PROSITE DATABASE	43
3.5 PROTEIN SIGNAL PEPTIDE AND SIGNALP.....	45
3.6 GENE ONTOLOGY MAPPING.....	50
4. SYSTEM DESIGN AND IMPLEMENTATION	52
4.1 SYSTEM MODELING AND APPLICATION ARCHITECTURE	52
4.1.1 <i>EST annotation system data modeling and data flow.....</i>	<i>52</i>
4.1.2 <i>Data presentation using web-based application</i>	<i>61</i>
4.2 SYSTEM CONSTRUCTION AND IMPLEMENTATION.....	62
4.2.1 <i>Parser development.....</i>	<i>62</i>
4.2.2 <i>Pipeline automation.....</i>	<i>66</i>
4.2.3 <i>Annotation web server design and implementation.....</i>	<i>68</i>
5. CONCLUSION AND POSSIBLE FUTURE EXTENSIONS	79

5.1 ANNOTATION OF PROTEIN DOMAINS USING INTERPROSCAN	80
5.2 GENE PATHWAY AND KEGG.....	81
6. REFERENCES.....	83
7. APPENDIX.....	87
8. GLOSSARY	94

TABLE OF FIGURES

Figure 1 Components of EST annotation system	3
Figure 2 DNA replication	7
Figure 3 Protein synthesis [20]	8
Figure 4 DNA sequencing [20].....	12
Figure 5 Major genomics data processing steps	14
Figure 6 GeneQuiz modules and control flow [2]	22
Figure 7 TIGR Gene indices page for <i>Aspergillus nidulans</i>	29
Figure 8 Categories of cDNA sequences [19]	37
Figure 9 A decision tree for prediction of cDNA full-length [19].....	38
Figure 10 Example of signalp results.....	47
Figure 11 Data modeling stages.....	53
Figure 12 System ER diagram	56
Figure 13 The data flow for annotation pipeline	60
Figure 14 BLAST results on annotation page.....	65
Figure 15 An example of a unigene annotation page	71
Figure 16 Search engines in annotation summary page.....	73
Figure 17 Predicted signal peptide in a protein sequence.....	77
Figure 18 Sequences aligned to form a contig.....	78

1. Introduction

1.1 Fungal genomics project overview

Naturally, fungi are very diverse, and fungal enzymes have the ability to perform chemically difficult reactions. This makes fungal enzymes potentially very useful in pollutant degrading as well as pulp and paper processing.

The fungal genomics project is funded by Genome Québec and Genome Canada. It aims to discover new and interesting enzymes produced by fungi. In this fungal genomics project, the researchers expect to identify over 70,000 new genes and aim to discover which of these are activated when exposed to various chemical substances. This is a typical genomics project that covers cDNA library construction, sequencing, microarray gene expression analysis, and detailed characterization of the target genes (enzyme encoding genes that have potential industrial applications).

The fungal genomics project is a multi-disciplinary project, bringing together the researchers in Concordia University's Center for Structural and Functional Genomics (CSFG) and experts from other disciplines. Those experts come from: the Biotechnology Research Institute of the National Research Council, the Pulp and Paper Research Institute of Canada (Paprican) and the INRS-Institut Armand-Frappier. By applying the most novel genomics approaches, with support from its internal bioinformatics group, the fungal genomics project presents an opportunity for discovering more useful fungal enzyme productions and improving the quality of the environment.

1.2 The statement of problems

In the fungal genomics project, researchers plan to study 14 fungal species. Thus, a large number of Expressed Sequence Tags (ESTs) are generated from each of the cDNA libraries. The raw sequence data initially consists of redundant, un-annotated information without any biological content. To further analyze the sequence data, we have built an automated sequence assembling and quality control pipeline, which integrates PHRED [9], LUCY [7] and PHRAP [12]. These three applications are used to process the sequence chromatogram files (both in ABI format and SCF format), remove low quality and vector containment sequences, assemble high quality sequences into consensus sequences, and provide feedback for individuals working in the labs. The feedback come in the form of quality reports for each batch of data collected from each species being examined [6].

Although the output from this pipeline can provide a first glimpse of these sequences, it tells us nothing about the structure or functions of these EST sequences. In order to support more thorough research in the fungal genomics project, such as, finding interesting target genes, microarray expression data analysis and enzymology classification, further analysis needs to be performed on the sequences.

To fulfill the mentioned tasks, two critical questions need to be answered by the bioinformaticians. First, from the biologists' point of view, what is the most useful information that can be derived from these sequences, and how can that information be obtained? Second, what is the most efficient way for the scientists to access this kind of information? Based on our research, all of these requirements can be accomplished by developing and implementing a well-designed EST annotation system.

1.3 Main achievement of our current works

At the time this thesis was written, we had finished the design and implementation of the EST annotation system, which included the following main components: the annotation pipeline, the annotation database and the annotation server [Figure 1]:

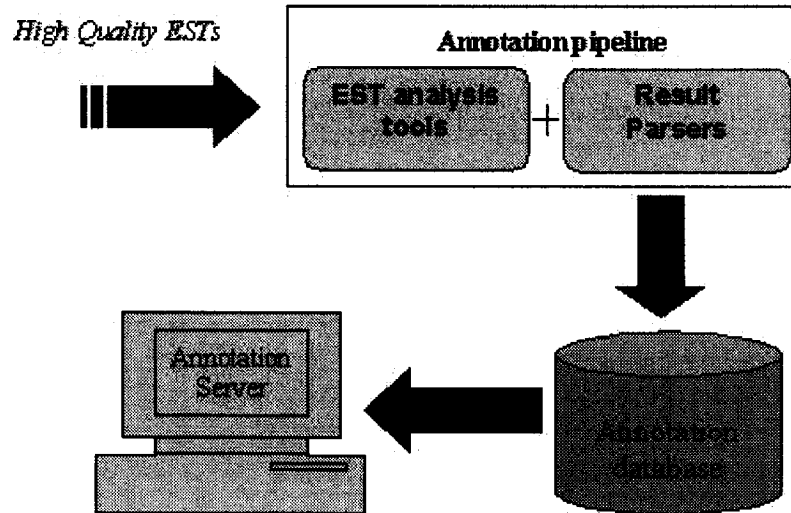


Figure 1 Components of EST annotation system

First, the annotation pipeline is a series of automatic processes, which integrates both public and in-house developed sequence analysis tools to further analyze the high quality EST sequences generated from sequence assembling and quality control pipeline [6]. It aims to retrieve as much biological information as possible for these EST sequences; in the meantime, it automatically parses the analyzed results using different parsers for easily loading the data into the annotation database. Second, a well-designed MySQL relational database stores all the annotation information generated by the annotation pipeline. The last component, a user-friendly web application called an

annotation server, is implemented to help users search and retrieve useful annotation information from the annotation database.

1.4 Contribution of this thesis

Based on the fungal genomics project, our in-house EST annotation system presents some different and interesting features when compare to other sequence annotation systems.

First, it provides a comprehensive data modeling in term of EST sequence analysis and annotation. It not only models sequence and some common features in a sequence annotation system, such as, sequence assembling information, sequence putative function, protein domain and motif, as well as GO designation, but it also includes important features that are not found in other EST annotation systems, including: sequence full-length prediction result, signal peptide prediction result.

Second, unlike other unigene-based EST annotation systems, our system also provides sequence-based annotation information and the links between these two datasets. Both of them are very useful references for further genomic research.

Third, to build a user-friendly annotation system, at the data analysis stage, we develop parsers to catch only the most interesting information from each of the sequence analysis results; at the system implementation stage, we provide users with a simple and effective way to access all kinds of annotation and reference information through well-designed search engines and annotation pages.

1.5 Layout of the thesis

Chapter 1 provides an overview of the fungal genomics project, outlines bioinformatics problems encountered in the project, and lists the main achievements elaborated upon in this thesis.

Chapter 2 presents the basic biological background knowledge required for this thesis and gives a brief comparison between some of the public EST analysis pipelines and our own data analysis pipeline. Following that is an evaluation of some existing sequence annotation systems and databases, which may be the most important references for our EST annotation system.

Chapter 3 discusses all the tools and methods that were implemented to analyze the sequence data in our EST annotation system.

Chapter 4 outlines, in detail, the design of our EST annotation system as well as implementation and development particulars.

Chapter 5 consists of my conclusions along with a discussion about possible future extensions.

2. Background

2.1 Biological conceptions and techniques

2.1.1 DNA, mRNA and cDNA

The first agent we need to know is **DNA**, a double-stranded molecule that encodes genetic information for any living organism [20]. DNA is composed of building blocks called nucleotides consisting of a deoxyribose sugar, a phosphate group, and one of four nucleotides containing the bases adenine (A), cytosine (C), guanine (G), and thymine (T). It forms a double helix structure.

Gene is the long polymer of DNA, which determines hereditary traits mainly through the following two processes: 1) DNA replication. 2) Protein synthesis.

DNA replication begins with a partial unwinding of its double helix, by using an enzyme known as DNA helicase. As the two DNA strands separate ("unzip") and the bases are exposed, the enzyme DNA polymerase moves into position at the point where synthesis will begin. Then the two original strands of the helix will do DNA synthesis separately according to the rules of complementary base pairing, linking C with G and A with T. This replication process is semiconservative so that, at the end of replication, each strand is paired with one of the newly synthesized strands. In this way, it produces two identical DNA molecules, each composed of one "old" and one "new" strand [22]. This process is demonstrated in [Figure 2]:

DNA REPLICATING ITSELF

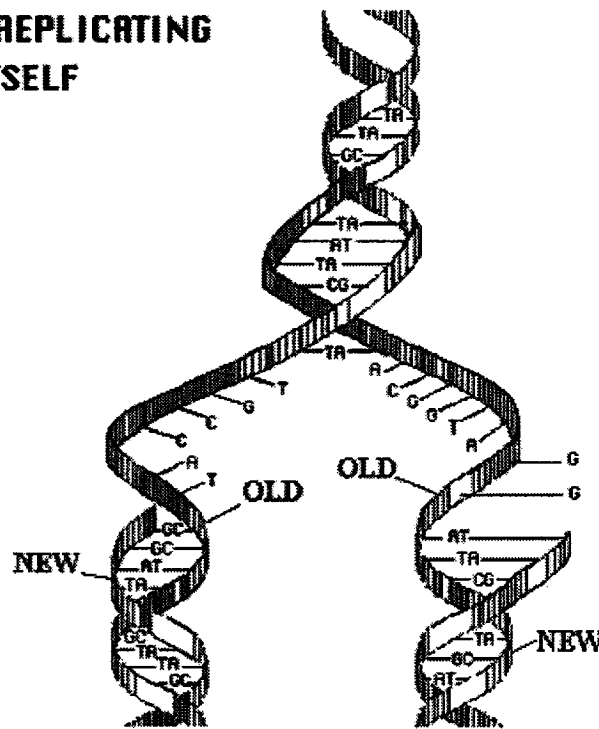


Figure 2 DNA replication

Protein synthesis is another important process for passing genetic information whereby DNA codes for the production of amino acids and proteins. Although important goals of many genomics projects may be to obtain genomic sequences and identify a complete set of genes, what scientists are really interested in are so called expressed genes. These genes are expressed as protein, a complex process comprised of two main steps.

Each gene (DNA) must be converted, or **transcribed**, into **messenger RNA** (*mRNA*), RNA that serves as a template for protein synthesis.

The resulting mRNA then guides the synthesis of a protein through a process called **translation**. Figure 3 shows the above processes.

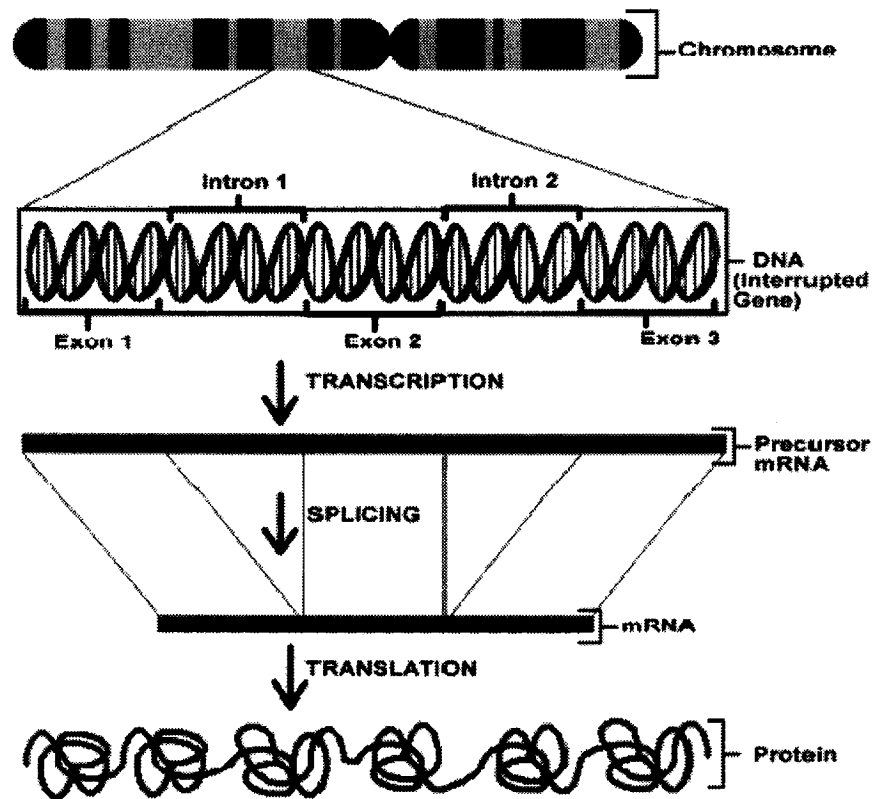


Figure 3 Protein synthesis [20]

A very important property for mRNA is that mRNA just includes the coding sequences (**exons**) of a gene, and does not contain sequences from the regions between genes (**introns**). Therefore, isolating mRNA is the key to finding expressed genes.

The main disadvantage of mRNA is that mRNA is very unstable outside of a cell; so, in an actual project, scientists use special enzymes to convert it to **complementary DNA (cDNA)**. cDNA is a much more stable compound and, by the way, because it was generated from a mRNA from which the introns had been removed, cDNA represents only expressed DNA sequences [20].

2.1.2 cDNA library construction, Cloning

To trace thousands of these expressed genes in the target organisms, in the fungal genomics project, biologists began their research by building a so-called **complementary DNA (cDNA) library** for each fungal species. A theoretically complete cDNA library could represent all of the genes that were expressed in a particular species. The biological term for creating such a cDNA library is called **cDNA library construction**. When constructing a cDNA library, each of these cDNAs is inserted into a cloning **vector** (e.g., a circular DNA plasmid), which has the capacity of self-replication and is replicated in a bacterium such as E.coli. This is the process called **cloning**. In the cloning process, vector replication occurs in such a manner that only one joined DNA molecule can propagate within a single bacterium. From a biologist's point of view, a population of bacteria containing a single inserted cDNA is called a **clone**. Clones can be cultured to make copies of the library for many experiments.

2.1.3 EST method and Sequencing

As we can imagine, using the traditional biological method to find an expressed gene in a cell is not that easy. Presently, thanks to the development of genomics, an emerging field for molecular biology and bioinformatics, a new and powerful method is being used frequently in different genomics projects for large-scale gene isolation. This new method is called the EST method.

ESTs (Expressed Sequence Tag) are tiny portions of an entire gene that can be used to identify unknown genes and to map their respective position within a genome. The

idea is to sequence bits of DNA that represent genes expressed in certain cells, tissues, or organs from different organisms and use these "tags" to fish a gene out of a portion of chromosomal DNA by matching base pairs [20].

EST data can be obtained by sequencing either one or both ends of cDNA, which represents an expressed gene. Sequencing only the beginning portion of the cDNA produces the so-called 5' EST; sequencing the ending portion of the cDNA molecule produces the so-called 3' EST. All of these ESTs are very useful in gene discovery.

DNA sequencing is the process for determining the order of the nucleotide bases along a DNA strand. This method is based on the principle that single-stranded DNA molecules that differ in length by just a single nucleotide can be separated from one another using polyacrylamide gel electrophoresis [20].

When sequencing is performed, first, the DNA to be sequenced is prepared as a single-stranded DNA called template DNA. Next, a short oligonucleotide is used as a primer, together with DNA polymerase and dNTPs, for the synthesis of a new DNA strand, which will be complementary to the template DNA. Then four DNA synthesis reactions are carried out synchronously by using different dideoxynucleotide (chain terminator), one for each of G, A, C and T. Every reaction includes the following components:

- A DNA template
- A primer
- DNA polymerase, an enzyme that drives the synthesis of DNA
- Four deoxynucleotides (G, A, C and T)
- One dideoxynucleotide: ddG, ddA, ddC or ddT.

The dideoxynucleotide is called a chain terminator because it lacks the necessary 3'-hydroxyl group. Whenever the dideoxynucleotide is incorporated into a growing DNA chain, DNA synthesis will stop.

After the first deoxynucleotide is added to the growing complement sequence, the synthesis reaction continues and adds base by base until a dideoxynucleotide is added that stops the process. Over a certain period of time, a series of DNA fragments of different lengths are generated from each reaction. All fragments from one reaction end with one same base (either G, A, C and T). Next, all four reaction mixtures are electrophoresed in parallel lanes in a high-resolution polyacrylamide gel under denaturing conditions. The resolution of the gel electrophoresis is so good that it can separate fragments differing in length by only one base. After electrophoresis, the final base at the end of each fragment can be identified directly by the sequencing machine, see Figure 4.

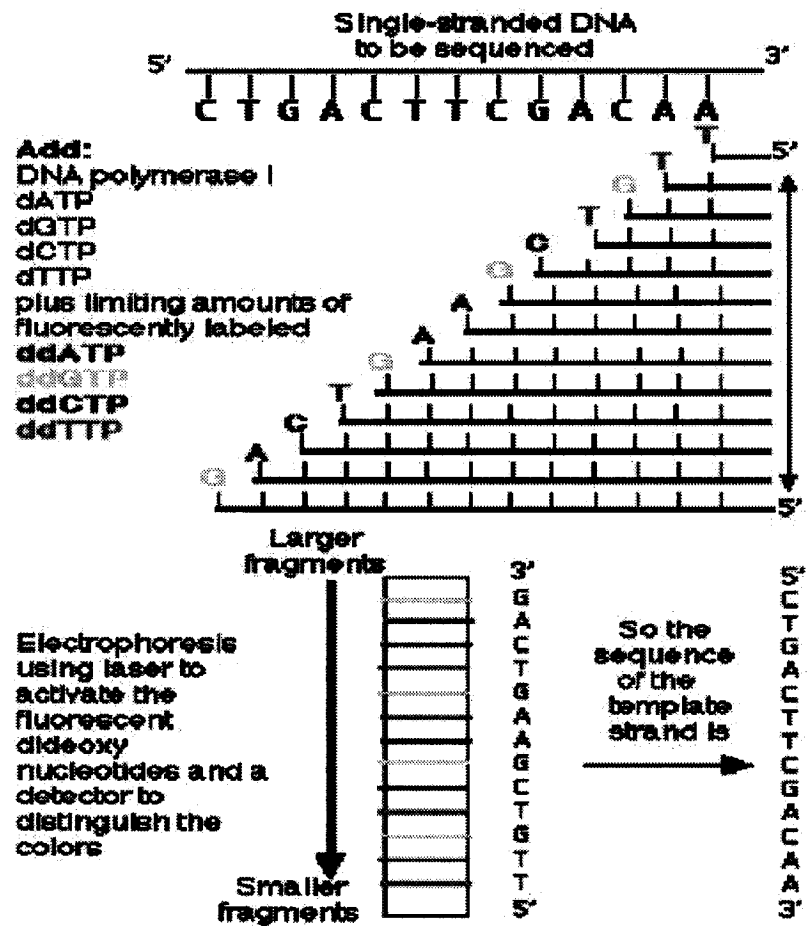


Figure 4 DNA sequencing [20]

In the Concordia fungal genomics project, researchers plan to identify interesting genes from 14 fungal species. Consequently, a large amount of individual EST sequence is sequenced from each cDNA library. Hence building an automated annotation system for function analysis and categorizing of these EST data has become an essential task for our bioinformatics group.

2.2 Existing EST analysis pipelines

2.2.1 Main steps of EST data analysis process

Figure 5 demonstrates the main steps of the EST data analysis process. In a typical EST project, to explore thousands of expressed genes in a target organism, biologists start their research by building a cDNA library, which theoretically represents all the genes that are expressed in that specific species. Subsequently, all the clones generated from this cDNA library are sequenced to determine the order of the nucleotide bases. The raw sequence data obtained directly from the sequencing machine is recorded in a series of chromatogram files, which can be processed by base-calling software. Base calling software is an automatic program that translates the sequence traces information into DNA sequence of bases. These DNA sequences can then be cleaned and assembled to form ESTs or unigene sequences for further genomics analysis.

In order to automatically analyze and transform large collections of raw DNA sequences generated by the biologists in fungal genomics project, we have built an automated sequence assembling and quality control pipeline, which integrates PHRED/LUCY/PHRAP to fulfill the tasks from step three to step five in Figure 5. Although, the output from this pipeline can give a brief idea of a given cDNA library, it tells nothing about the structure and function of these sequences. Neither does the output gives scientists any idea of genes they are interested in, hence more sequence analysis needs to be performed for further genomics research.

If no annotation database exists, scientists normally use the unigene (or EST) dataset generated from sequence assembling program to do homology search against

NCBI (National Center for Biotechnology Information) non-redundant databases or their own target databases. Afterward they parse the results using some open source programs or their own tools to find putative functions of these sequences or their interested target genes in the cDNA library. However, because of the limitation of information that the sequence homology searching can provide as well as the variety of requirements from different EST projects, the results obtained from the above sequence analysis processes are far from enough for an actual EST project. Thus, building a sequence annotation system for sequence function assignment, data storage, and visualization are becoming more and more essential for most EST projects today.

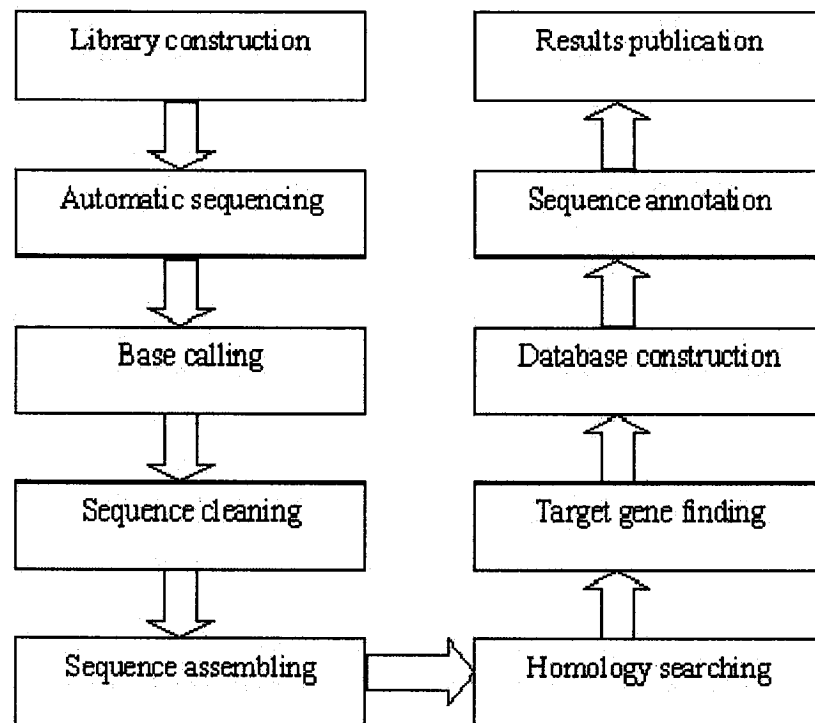


Figure 5 Major genomics data processing steps

2.2.2 Existing EST analysis pipelines

It is known that most genomics studies are based on the basic assumption that the sequences being used are trustworthy. However, the raw sequences coming directly from an automatic machine may not be reliable for the following reasons: the sequences might contain errors, the Dye-terminator reaction might not have happened, or short insert, vector contaminant sequences and low quality sequences could potentially be mixed in the high quality sequences and so on.

To obtain the clean high quality sequences and assemble them into longer DNA pieces for further genomic data analysis, an automated sequence processing and analysis pipeline is very essential. Recently, some research groups have published their automated procedures for EST data analysis. The following gives a brief overview of some of these systems for users to get a general idea of what can be done in this stage:

The PipeOnline system, implemented by the OSU Bioinformatics Group at Oklahoma State University is a series of integrated and automated sequence analysis programs, which can be used to process large collections of raw DNA sequence data from chromatograms or FASTA files and associate function at the gene level [4].

PipeOnline begins by reading input from chromatogram files either in Applied Biosystems Inc. (ABI) format or Standard Chromatogram Format (SCF) format. It then uses PHRED as base-caller, and converts them to bases and quality indices. The program called CROSSMATCH is used to compare FASTA sequences from the PHRED results with the vector sequence, and finally the sequence assembler called PHRAP is used to construct consensus sequences (or contigs) from the input files. To get a cleaned unigene

dataset, a Perl script called `removex.pl` is used to clean up the vector sequences from the singleton sequence files after the assembling.

In addition to performing the above sequence analysis tasks, PipeOnline also provides a basic function assignment, associating DNA sequences with gene functions, through protein database searching and functional sorting. By running NCBI BLASTX in batch mode, the processed output DNA sequences are compared with public non-redundant amino acid databases to find sequence similarity, and the top five significant alignments are stored in the PipeOnline database. For each record, the system performs an automated functional sorting based on the Metabolic Pathways Database functional dictionary. All matches between NCBI protein records and the MPW functional dictionary are stored in a MySQL database table to generate a NCBI protein-function dictionary (a gene index numbers to MPW function matching table), which can be queried and sorted locally.

The EST Pipeline System, designed and implemented by the HangZhou Genomics Institute, at ZheJiang University is another typical EST analysis pipeline. This is a highly automatic data analysis pipeline designed for EST projects. It integrates web interfaces, background Perl scripts and third-party bioinformatics tools and databases to complete the sequence analysis tasks, including: base-calling, screening, assembling and functional classification work.

This pipeline can accept chromatogram files, sequence files plus quality files or just sequence files as input. To prepare cleaned sequence data for assembling, it uses CROSSMATCH to mask vector and provides options for users to mask other sequences

such as: E.coli contaminants, low complexity sequences. It also filters out short sequences that are shorter than 100 bp. In the last step, the cleaned ESTs are assembled using either PHRAP or CAP3 (by Huang, X. and Madan, A.) according to the users' preference [28].

In addition to providing the automatic processing of EST sequence raw data, this pipeline also generates some additional features for basic gene annotation and classification. It uses BLAST to check contigs against the NCBI non-redundant nucleotide and protein databases and the SWISS-PROT protein database to find homologies. Furthermore it performs HMM Pfam search to find potential protein domains, and it also implements a GOTREE module to draw a tree view of the contigs according to their annotation and their GO id assignment. The connected list of views presented in the report pages makes this system very user-friendly and effective for accessing the analysis results.

The next system we will look at is called ESTAP (EST Analysis Pipeline), a co-development software project supported by the Samuel Roberts Noble Foundation in Ardmore Oklahoma, the University of Nevada at Reno and the Virginia Bioinformatics Institute. ESTAP is an automated EST processing package for EST sequence cleaning, similarity search, and assembling [17].

The first step of ESTAP is the sequence data validation process. In this stage, the system automatically reads input EST sequences, checks the integrity of the raw data and loads them into database. Subsequently, it cleans the raw sequences by removing low quality end sequences, vectors, polyA (or polyT) chains, chimera and contaminations

from cloning vector, the host, or user defined organisms. The cleaned sequences are then compared with DNA or protein databases using the BLAST algorithms to perform similarity searches. ESTAP also clusters, using `d2_cluster` (Hide W. *et al.*), and assembles (using CAP3) ESTs from the same cloning library to form a unigene set. Finally, all the unigene dataset are automatically annotated using the InterProScan program from EBI (European Bioinformatics Institute)

EST-PAGE is another public EST sequence management and analysis system, implemented and used in a joint project between the Bioinformatics and Computational Biology and the Bovine Functional Genomics Laboratory, funded by ARS, USDA.

PAGE is the acronym corresponding to the following EST data processing steps: P is the processing of raw chromatogram files for base calling using PHRED; A stands for data analysis of sequence data to screen out vector and low complexity sequences and check for Ecoli contamination; G is the process of preparing quality sequences to submit to GenBank dbEST; E represents exploration of the EST data to find redundancy in plates or in the library, and sequence clustering (or assembling) using CAP3 [18].

In addition, EST-PAGE handles a variety of aspects of an EST project, such as, clone tracking, library statistics, BLAST search, and GO (gene ontology) association, through its web interface.

The last EST analysis system we will review is the ESTAnnotator implemented by the HUSAR Bioinformatics Lab at the German Cancer Research Center. Although the

name of this system is called ESTAnnotator, it is actually a high throughput EST process pipeline [13].

When processing the EST data, ESTAnnotator also uses PHRED to do the base calling. The program called Repeatmasker is used for masking either repeats or vector sequences. The difference between ESTAnnotator and the previously mentioned systems is: before assembling ESTs into consensus sequences, it runs a BLASTN search against an organism-specific database, and only sequences with hits with an minimum expectation value (less than e^{-20}) are included in the assembly. In the last step, the BLAST analysis is performed again at the protein level: at this time, BLASTX is run against the SWISS-PROT database and TBLASTX search is performed against EMBL ESTs. The following Table 1 is the summary of these different pipeline systems:

Table 1 Summarizing and Comparison of different EST pipeline systems

System	Literature reference	Analysis Tools used	Additional features
PipeOnline	PipeOnline 2.0:automated EST processing and functional data sorting [4]	Phred/Crossmatch/Phrap	BLAST search Functional sorting
EST pipeline system	EST Pipeline System: Detailed and Automated EST Data Processing and Mining [28]	Phred/Crossmatch/Phrap	BLAST search Protein domain GO classification
ESTAP	ESTAP - an automated system for the analysis of EST data [17]	D2_cluster, CAP3	BLAST search
EST-PAGE	EST-PAGE: managing and analyzing EST data[18]	Phred/Crossmatch/CAP3	BLAST search Submission to dbEST
ESTAnnotator	ESTAnnotator: a tool for high throughput EST annotation[13]	Phred/RepeatMasker/CAP	BLAST search
Our sequence analysis and quality pipeline	https://fungalgenomics.concordia.ca/internal/activities/doc_index.html	Phred/Lucy/Phrap	No

From the above we can see that: although the sequence analysis tools selected by each system may be slightly different, most of EST analysis pipelines share the following common features:

- Raw sequence base calling and quality assignment
- Sequence cleaning to remove low quality, low complexity, vector and other contaminated sequences
- Sequence assembling or clustering to form contigs (or unigenes)

Most of these EST analysis pipelines provide some additional features to annotate their data, such as, searching for sequence similarity with known genes deposited in public databases to assign functions. However, in terms of fully annotating large EST datasets to support further genomics research, none of these systems can provide enough annotation information. Therefore, it is necessary for us to survey some more comprehensive sequence annotation systems to discover the essential and most important features encoded in these sequences, and to report back the details in which the scientists have an interest.

2.3 Sequence annotation system overview

2.3.1 Main goals of sequence annotation system

Since the information held in a genomic sequence is encoded and highly compressed, in order to extract biologically interesting data we must decrypt this primary data computationally. In the area of genomic research, it is increasingly accepted that an annotation system for these sequences can greatly enhance their biological value.

Building a sequence annotation system involves many processes of adding biological information to a sequence, which are very complex and always evolving tasks.

In a genomic project, a sequence annotation system can be implemented either on genome sequences or on EST sequences, in accordance with different research purposes. Regardless of which category the sequence data belongs to, the basic approaches of building sequence annotation systems are the same. They include integrating different sequence analysis tools and automatically parsing the results, finally presenting them in a user-friendly manner.

A genome sequence annotation system annotates the whole genome sequence, which consists of large pieces of chromosome. Its information may include possible gene location, introns, exons, homologies to known genes, gene mapping, and gene signals such as promoters and splice sites.

An EST sequence annotation system mainly focuses on describing the structure and functions for an expressed gene within a genome. It may include a full-length prediction, a putative function assignment, homologies to known genes, the results of a function domain search, a protein signal peptide prediction and other useful features.

In the past, most of the sequence data has been annotated manually. This approach of annotation is very useful, and will continue to be; however, the volume of the sequence dataset has accelerated to the point that it is impossible to have sufficient curators to annotate every new sequence all by hand. Thus, for most genomics projects, an automated approach for sequence annotation has become very essential.

2.3.2 Existing sequence annotation systems

A number of researchers from major genomic centers have implemented sequence annotation systems for different purposes. An evaluation of these systems can give a brief

idea of how sequences are analyzed and annotated, and help our bioinformaticians to fully understand the roles and main goals for sequence annotation systems in genomic research. Here we discuss three of the major sequence annotation systems to reveal their important features, limitations and their influence on our own EST annotation system.

2.3.2.1 GeneQuiz system

GeneQuiz from EBI (European Bioinformatics Institute) is a typical sequence analysis and annotation system. It integrates a variety of search and analysis tools and up-to-date nucleotide and protein databases, creating a compact summary of findings and presents the results through a web-based browser [2]. GeneQuiz takes a protein sequence as input data and generates specific function classification and annotation for this sequence as an output. It consists of four modules: GQupdate, GQsearch, GQreason, and GQbrowse. The Figure 6 shows the dataflow for the GeneQuiz system:

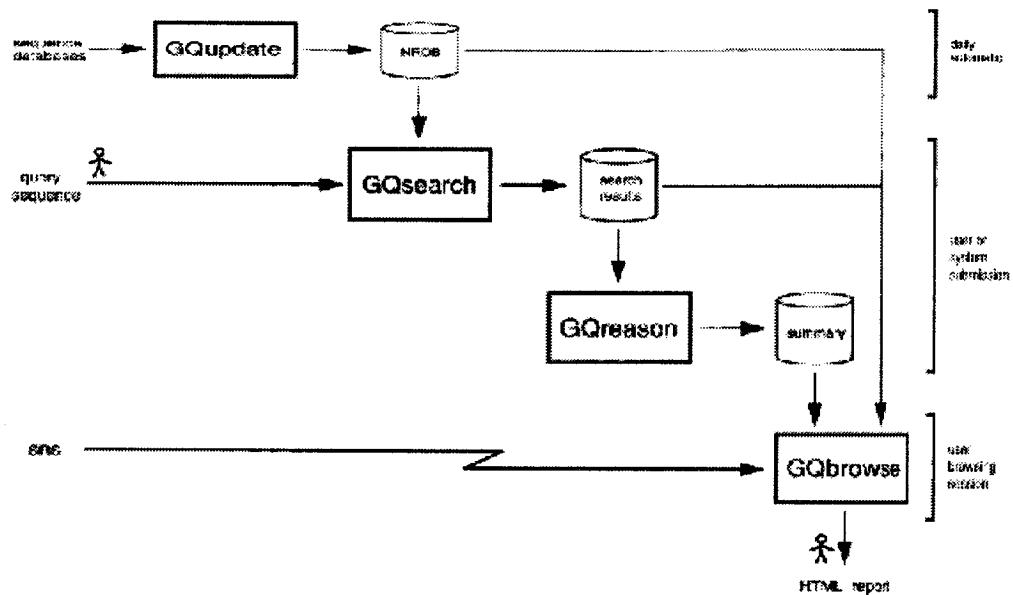


Figure 6 GeneQuiz modules and control flow [2]

GQupdate, the first module of GeneQuiz system, is in charge of integrating and updating non-redundant nucleotide, protein sequence databases, and databases of protein structure and motifs from public domain. To obtain the utmost information from various biological databases, the GQupdate module defines a configuration file containing a series of Internet address of these database entries and paths to target files. It then runs as an autonomous module on a daily basis to get the updated version of these databases via ftp. After updating the databases, it automatically performs all the necessary reformatting procedures to make the databases searchable by the various search programs.

GQsearch performs protein function analysis using a collection of mostly standard publicly available sequence analysis programs. These analyses include sequence homologue search by similarity, detection of motifs and the conservation patterns in the corresponding protein family and predictions of secondary and tertiary structure.

GQreason provides both general functional classification and specific functional annotation by carefully choosing homologues and systematic analysis of the sequence database annotations. The method used by GQreason for general functional classification is based on the generation of a keyword/class association dictionary. Assignment of a class to a new sequence is determined by looking up the keywords for that sequence in the dictionary and assigning the most frequently associated class. Specific functional annotation of a sequence is performed by a lexical analysis procedure to its sequence database description fields acquired from the homologue search. The lexical analysis consists of a series of comparison tests using regular expressions and key words. The detailed description of the algorithm can be found in [2].

GQbrowse, the last module of the GeneQuiz system, allows users to view the final results through a web interface, and links to external public sequence and pattern databases via SRS (Sequence Retrieval System).

GeneQuiz provides a good protocol for integrating different sequence analysis tools to build the annotation pipeline. However, it is not a complete genomic sequence annotation system to address problems in an actual genomic project because of the following limitations:

- The main focus of this system is for protein sequence function analysis and assignment. It lacks the ability to handle large-scale raw DNA sequence analysis.
- GeneQuiz does not provide a complete sequence annotation database to record all the annotation information generated in each analysis step. It only provides tables to store the final analysis results for visualization.

2.3.2.2 Ensembl genome sequence annotation system

Currently many bioinformatics researchers are moving towards developing a comprehensive genomic sequence annotation system, ranging in focus among sequence analysis, data storage and visualization. An example is the Ensembl system developed by the Wellcome Trust Sanger Institute and European Bioinformatics Institute (EMBL-EBI)[14].

Ensembl is a carefully developed software engineering framework that aims to provide a bioinformatics system that is easy to apply to different organisms and data types. Ensembl consists of three main portions: the Data analysis pipeline, Ensembl genome annotation and the Ensembl web site.

The Ensembl analysis pipeline system is designed to perform a set of sequence analyses on the entities stored in the database.

Since the results of one analysis may be the input for other analyses, there is a need for ordering the sequence in which analyses are done. In the Ensembl system, RuleManager is used to accomplish this task. To make it work, a series of rules is used to define the dependencies among analyses. Once the system has determined that a given analysis applies to an entity, it finds all the rules associated with this analysis and checks to make sure that all pre-requisite analyses specified in the rules' conditions have been run and completed. If the requirements are met, then the RuleManager will trigger the identified goal analysis to run. The Ensembl analysis components are designed in a very flexible way, which allows the developer to write new components rapidly and construct composite processes.

Ensembl genome annotation is the most crucial part of the Ensembl system for the annotation of known genes and prediction of novel genes. The Ensembl gene build system incorporates a wide range of methods including ab initio prediction (Genscan), homology and gene prediction HMMs (Hidden Markov Models) [14]. Genscan studies each DNA sequence and identifies DNA regions that may be genes (exons). Exons from these predictions are then compared to the sequences of all known genes in the public databases in order to provide “supporting evidence” using BLAST. All these “Ensembl genes” are stored in the databases for later use.

The Ensembl web site provides a variety of alternate views of the data, including: Chromosome map, Contig view, Gene view, and Protein view. Ensembl also adopts the

DAS (<http://www.biodas.org>) standard, which enables users to easily view and compare annotation from different sources distributed across the Internet.

Ensembl contains the most important features required for a genome annotation system. However, in terms of performing the analysis and annotation on large-scale EST dataset, it presents the following limitations:

- Ensembl models the genomics sequence at the genome level rather than EST level, it lacks the methods and database schemas to directly handle the raw data obtained from an EST project.
- Although Ensembl is being continuously refined and expanded to include more new data types. However, due to its complex system hierarchy and data structure, directly adopting it in its current form, for both EST analysis and annotation, is not a simpler task than implementing an in-house EST annotation system.

2.3.2.3 TIGR Gene indices

The TIGR Gene indices are a collection of species-specific databases that use a highly refined protocol to analyze EST sequences in an attempt to identify the genes represented by that data and to provide additional information regarding those genes [23]. The basic idea behind the construction of TIGR Gene consists of assembling the public ESTs from GenBank to generate a high-fidelity set of non-redundant transcripts (called TC or “tentative consensus”), which can then be used for more extensive functional annotation. The following are the main steps of this process:

The first step is the construction of a database of annotated gene sequences. For each species-specific Gene Index, all sequence records from GenBank are downloaded as

well as CDS join features for full-length genes and mRNA sequences. The records are then parsed [24]. These sequences are then cleaned in order to identify and remove contamination sequences (such as vector, adapter).

In the second step, cleaned ESTs, ET sequences (expressed transcript) from TIGR EGAD (Expressed Gene Anatomy Database), and TC sequences from the previous build are compared to identify overlaps and to be grouped into clusters.

In the last step, all these ESTs, ET sequences and TCs in a cluster are assembled using CAP3. The resulting set of TCs is loaded into the TIGR Gene indices database for annotation.

After the assembly, the TC sequences are annotated for functional assignment. The strategy used for TIGR Gene indices functional annotation is not very complex. The main point is using consensus sequences (TCs) rather than ESTs to represent each gene, as consensus sequences are longer and more likely to contain protein-coding sequences than individual ESTs. A TC sequence contains ET sequences (or is formed from previous TCs) with an assigned gene assigned the function of that gene. TCs without function assignment are first searched against protein databases and then against nucleotide databases to find high-score hits and retrieve putative functional annotation.

The following two important features make TIGR Gene indices more user-friendly and easy to use:

First, TIGR Gene indices maintain heritability, which is the ability to effectively track assemblies across releases. New assemblies are assigned a new and unique TC identifier. Previously used identifiers are kept to allow for tracing back. Second, TIGR Gene indices provide different ways to access EST annotation. For example, Gene

indices can be searched by TC number, ET id, GenBank accession number or gene name.

Figure 7 is an example of TIGR Gene indices for *Aspergillus nidulans*

Tigr Gene indices provide a good example for EST analysis and annotation in term of EST assembling, function assignment and search engine design. The research goals of Tigr Gene indices are close to what we have for our EST annotation system. However, it still lacks the following important features for supporting an actual EST project:

- First, the data source of Tigr Gene indices are the ESTs deposited in the public domain, it does not (and there is no need for it to) provide the linking between the unigene sequences and ESTs, which information is very important in an actual EST project.
- Second, it lacks the important information to support downstream search works: such as, sequence full-length prediction, signal peptide prediction, GO designation.

easier and more straight-forward way than modifying and adopting other systems to address our problems.

Table 2 is the summary of the main features of the above annotation systems:

Table 2 Main features of GeneQuiz, Ensembl and Tigr Gene indices system

	GeneQuiz	Ensembl	TIGR Gene indices
Genome annotation Or EST annotation	Neither	Genome annotation system	EST annotation system
Performing different sequence analysis tasks	Yes	Yes	No
Providing functional assignment for sequence	Yes	Yes	Yes
Data presenting methods	Web browser	Multiple viewers	Web browser with search engines
Local annotation database	No	Yes	Yes
Adopted to our EST data	Cannot	Difficult	Cannot

2.3.2.4 Data modeling in major annotation databases

The last part of this chapter gives a brief overview of some major public annotation databases, including: Ensembl, PEDANT, PANTHER, and SGD (Saccharomyces cerevisiae genome database), to reveal what kinds of sequence-related information are modeled in these systems. This can help us to gain a deeper understanding of the roles a sequence annotation system plays in genomics research.

The main focus of Ensembl genome database is to track all the annotation information around the sequences of large genomes that are generated by the Ensembl sequence analysis system. This includes the information about genome sequences, genes (known genes or genes predicted by the Ensembl system) and other interesting features.

The data modeled in Ensembl system can be logically separated into two main groups [8]:

The first group is biologically meaningful objects, including: chromosome, contig, gene, exon, and transcript. Chromosome describes only real chromosomes or alternative sequences of reference chromosomes; Contig (or consensus sequence) serves as basic sequence holder for genome sequences; Gene is used to record information for known genes or genes predicted by the Ensembl system; Exon stores data about exons, associated with transcripts via exon transcript; Transcript is derived data from exon, and used to retrieve and storage sequence transcript region.

The second group is features and other metadata, including: features, mapping information, external reference and other objects. Features are a series of objects used to describe analysis programs and features created by these programs for a piece of sequence, such as, repeat regions, DNA alignment information by similarity, features on the translations and so on; Mapping information are used to record mapping identifiers between different releases; External reference is a table designed to store links or descriptions of external references to other databases; Other objects model and record annotation information that does not fit into the above categories.

PEDANT (Protein Extraction, Description, and Analysis Tool), from the Munich Information Center for Protein Sequences (MIPs) is another high-throughput genome sequence analysis and annotation system.

The PEDANT genome database is designed to store, modify and access well-organized annotation information on completely sequenced and unfinished genomes [11].

PEDANT is based on a standard RDBMS and the SQL language. There are two major types of SQL tables in PEDANT database. The primary tables are used to store sequence raw data, such as, genomes, contigs, ESTs, exons, genes, ORFs, proteins, as well as the results from different sequence analysis tools, for example, BLAST output, Pfam results. These results are subsequently parsed and then stored in the secondary tables. The secondary tables are mainly designed to save the properties of the DNA sequence, genetic elements and individual pieces of evidence derived from different analysis processes.

Because of the open architecture of PEDANT system, novel analysis tools and results can be easily adopted to the PEDANT database without major changes in its data model.

PANTHER from CELERA Genomics is another type of sequence annotation database that is specially designed for characterizing the function of proteins on a large scale. The most important feature of the PANTHER system is to provide a simple ontology of protein function by associating the ontology terms with groups of protein sequences rather than individual sequences [26]. By the way, by using a set of “training sequences” to build statistical models (Hidden Markov Models), the system can also be used to accurately classify novel protein sequences.

PANTHER is composed of two main components: the PANTHER library (PANTHER/LIB) and PANTHER index (PANTHER/X). The PANTHER/LIB is a collection of data models representing protein family, subfamily, family tree, statistical model and related sequences; while PANTHER/X consists of data models to classify a gene into different categories by ontology, such as molecular function and biological process [26].

Although PANTHER is not a true genomic sequence annotation system that can be easily adopted by an actual genomic research project, it is still a very useful reference for large-scale protein sequence function assignment and classification.

The last genome database discussed in this chapter is SGD (Saccharomyces cerevisiae genome database), funded by the National Human Genome Research Institute at the US National Institutes of Health. It provides a well-organized collection of biological and genetic information about the model yeast organism *Saccharomyces cerevisiae*. SGD implements comprehensive data modeling for *Saccharomyces cerevisiae*, ranging in focus from the whole genomic sequence, its genes and products, its mutant phenotypes, and all the literature supporting this data [25]. The data modeled in SGD can be roughly grouped into four categories. The first category models genetic information for the whole genome, including chromosome, clone, DNA sequence, protein, and locus related information. The locus in SGD is used for storing genetic information, normally is a gene name, characterized by a mutant phenotype or by a DNA sequence. Locus related tables contain information for SGD locus, gene products, and gene reservation. The second category is the feature and reference information. The feature consists of information about sequence features, and sub-features that are found in regions of sequences, for example, ORFs, tRNAs or genes has been characterized. The reference contains all the reference information for the data in SGD, which can be from a book, journal, or personal communication. The third category is the additional features, such as, sequence homology, Gene Ontology, mutant phenotype, taxonomy (derived from the NCBI). The last category is the miscellaneous information that is not directly related to

genome sequence, but still very useful in SGD system, such as, administration information, curator tables.

To summarize, although the data modeled in the above annotation databases vary according to different research purposes, they all cover two main aspects of genomic sequence, which are sequence and interesting features. Since the main purpose of building an EST annotation system in fungal genomics project is to present EST data and their interesting features to support fungal genomics research, the data modeling strategies used in the above systems may serve as useful references in the system design stage; however, the most challenging task for the designer of this system is to dig out what are the most useful features encoded in these EST sequences, and how to get them. In the next chapter, we will discuss about this question in depth.

2.3.3 Basic requirement for presenting our EST data

The following summarizes the basic requirements for our EST annotation system, as proposed by the biological researchers at the initial stage of the fungal genomics project. Although the actual system may include more functional assignment, it still serves as a starting point for our system design.

Annotation should be implemented both on assembled sequences and EST sequences, with each annotation page including the following information:

- Sequence id with its nucleotide sequence
- Predicted translated amino acid sequence and its frame value
- For consensus sequences: show sequences and other members within the contig along with links to their individual details

- TargetFinder results
- Similarity search (BLAST) results
- Motif search results
- Protein localization signals (secreted, nuclear targeting, mitochondria) i.e., SingalP results.
- Gene ontology designation

To obtain the above information about our sequence data, we should provide a flexible, transparent system to automatically run a series of sequence analysis programs on these sequences and present the results in a user-friendly way. The descriptions of all the related methods are in the next section.

3. Methods used

3.1 Annotator: A basic sequence annotation program

The first method we applied to analyze the EST data is Annotator, a program developed by Dr. X.J. Min, a member of our bioinformatics group. Annotator was designed to perform both EST cDNA full-length prediction and the functional annotation of sequences. The program uses the output of BLASTX (a sequence analysis method described in the next section) to assign a function to a query sequence, and to predict whether or not a query sequence includes an intact ORF (open reading frame). It accomplishes this by searching for putative start codon and stop codon within an ORF [19].

The usage of sequence similarity for the functional annotation of ESTs (or cDNAs) is a widely accepted practice amongst major genomics organizations. BLAST from NCBI is the most commonly used program for searching nucleotide and protein databases, in order to find sequences that are similar to a given query sequence. BLASTX (which compares a nucleotide query sequence against a protein database) is able to reliably identify protein coding regions of a DNA query by performing a database similarity search. In the Annotator program, the author presents a new, practical algorithm to combine functional annotation and full-length prediction, the results of which form the basis of our annotation system. Furthermore, the program predicts whether or not the protein coding region of a query sequence is completely determined [19].

Theoretically, a typical full-length cDNA sequence in eukaryotes contains a 5' untranslated region (5' UTR), an amino acid coding region or an open reading frame

(ORF), and a 3' untranslated region (3' UTR) [19]. An ORF is marked by the presence of a start codon (ATG) in the 5' region and a stop codon (TAA, TAG, or TGA) in the 3' region. Hence all cDNA sequences may be classified into five categories: Full-length, short full-length, possible full-length, ambiguous and partial. See Figure 8 for more details:

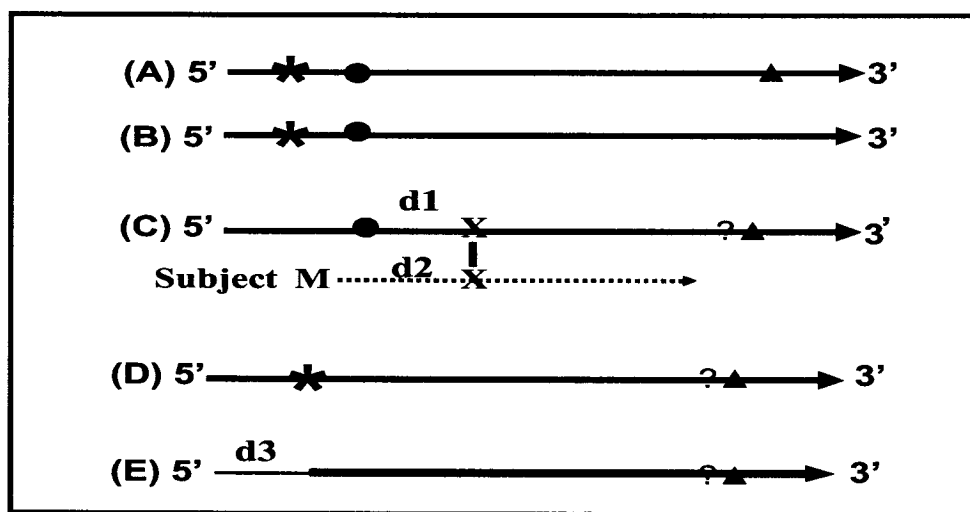


Figure 8 Categories of cDNA sequences [19]

Thick solid line: cDNA sequences after processing by Lucy
Thin solid line: a low quality region of a cDNA sequence removed by Lucy at 5' end
Dashed line: amino acid sequence of the subject in BLASTX
*****: Stop codon before start codon (5' end stop codon)
●: Predicted start codon within an open reading frame
▲: Stop codon after start codon (3' end stop codon)
?: Indicates checking if a 3' stop codon exists
X: The first amino acid in the alignment of the highest score pair in BLASTX
M: Methionine
d1: The length of predicted peptide from a predicted starting codon to X
d2: The length of M to X in the subject sequence of the highest score pair in BLASTX
d3: Length of cDNA sequence trimmed by Lucy including a portion of a vector, an adaptor and a low quality region of a cDNA insert

This classification is important to keep in mind in so far as: by searching for a start codon, the 5' start codon and the 3' stop codon, we are able to predict the full-length

status of a given sequence. By searching within the reading frame for the 3' end stop codon, as well as considering the position of the 3' stop codon in the query sequence relative to the subject sequence length in the BLASTX alignment, we can determine the complete status of a cDNA coding region. Figure 9 shows the full-length prediction algorithm:

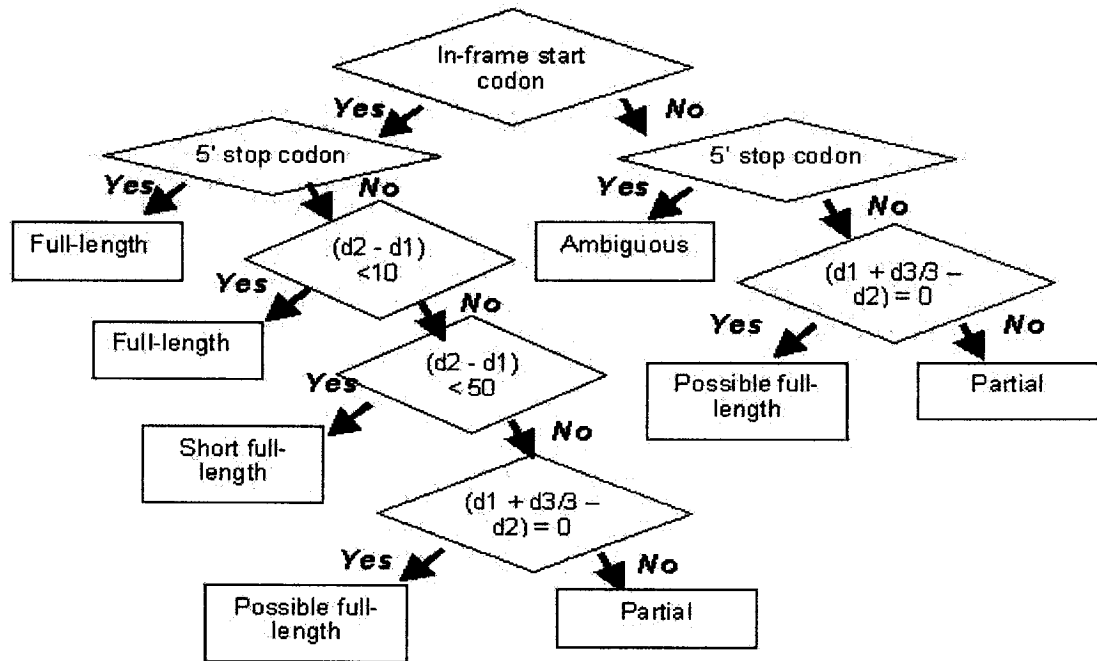


Figure 9 A decision tree for prediction of cDNA full-length [19]

To predict whether the 3' coding region is completely obtained, if there is a stop codon at the 3' end and:

$subject\ sequence\ length - subject\ beginning\ position\ in\ the\ alignment - (query\ sequence\ length - query\ beginning\ position\ in\ the\ alignment) / 3 \leq 0,$

then the 3' coding region of the query sequence is completely obtained. Detailed explanation of this algorithm can be found in [19].

Annotator produces a tab delimited text file that can be easily loaded into a relational database (RDB) allowing for easy access to and further usage of its results. From a biologist's point of view, the results from this method provide a brief overview of the status of all the EST sequences, and may later be extremely helpful when characterizing the novel genes through laboratory experiments.

3.2 Sequence frames and translation

In order to obtain a more in depth analysis, as well as functional annotation results from our EST data, translation of all these nucleotide sequences to protein sequences is an essential step. Sequence translation is a cellular process by which protein synthesis from RNA is achieved. Once the RNA has been transcribed, it travels from the DNA template to the ribosome on the endoplasmic reticulum to be translated for protein synthesis. Each 3 bases in the RNA sequence codes for 1 amino acid.

The translation might start from one of 3 positions, from either end of a given sequence (forward or reverse). Therefore there are always 6 possible predicted protein sequences, referred to as the six possible reading frames.

The algorithm used for sequence translation is not very complex and there are many programs available on the Internet to perform this task, for datasets of any size. However, the majority of these tools cannot predict which reading frame should be used for a specific sequence, so they either prompt the user to determine the reading frame or simply iterate all the six possible frames of translations. From a practical perspective, these tools are not very useful in our project as we wish to analyze thousands of unknown

sequences together. For this reason, another program, Translator, was developed by Dr. X.J. Min to address this problem.

Translator is a Perl script that processes the BLASTX output from the large sequence file and then translates the nucleotide sequences into protein sequences. For each of the queried sequences with a BLASTX hit, the frame value from the BLASTX result is retrieved and the original sequence is translated into a protein sequence in the same frame. The output of this program is a FASTA file, which contains the results of all the translations, facilitating direct use of the output by other protein analysis tools. Another advantage of Translator is that the peptide mass has been calculated and added into the translation results for all the assembled sequences. This proves to be a very useful reference value for these sequences. The following lists some sample Translator output:

> Contig2000 translated amino acid sequence (frame +3; peptide mass 48472)

```
LLFPSSPFRSFLVAFLPAFFKFPFSSHSFRPSFSSLASAVLLNHPI SGAVCIFHSHTFSVISLTGPDTHSL S*YHLP  
SILSPTSSSTMKGTAVATALALGASTALAAPSSTIKARDDVTAITVKGNAFFKGGDRFYIRGVYDYPGGSSKLADPIAD  
ADGCKRDIEKFELGLNIRVYSVDNSKDHDECMNALADAGIYLVLDVNTPKYSLNRADPAPSYNDVYLQYIFATVDKF  
ASYKNTLAFFSGNEVINDGPSSKAAPYKAVTRDLRQYIRSRNYREIPVGYSAADIDTNRLQMAEYMNCGTDDERSDF  
AFNDYSWCDPSSFTTSGWDQVKNFYGLPLFLSEYGCNTNKREFEEVSALYDTKMTGVYSGGLVYEQESSDYGLV  
EINGDSVKTLSDYDALKSAYSKTSNPEGDGGYNTGGANPCPAKDSPNWDVDGDSLPAIPEPAKKYMTGAGKAGFSG  
SGSMNAGTASTSTATPGSGSASSSSSSSGSSGTSTSSSTGAAAGLQVPGFAMAPVMVGLVTVLSTVFGAGLVLL *GRRVV  
WMIFCIFPCVGLRCVSFPLFLFFSFSSFCLLPEGGFK*CRVGLGAKQHSDMCTVGLDHP*PVPVDV*SI***TEPR*H  
PCFFQKKKKKKK
```

3.3 NCBI BLAST for sequence similarity

As previously mentioned, BLAST (Basic Local Alignment Search Tool) programs are widely used for searching nucleotide and protein databases for sequence similarity to a target sequence, in order to develop a hypothesis concerning relatives and functions. The following is a brief summary of this method and how it was implemented in our existing data analysis processes:

Introduced in 1990, BLAST programs consist of a set of sequence comparison algorithms, which are used to search sequence databases for optimal local alignments to a query sequence. The term local alignment refers to the alignment of some portion of two nucleic acid or protein sequences.

While performing comparisons, BLAST takes a query sequence and aligns it with each sequence in the database in a pair-wise fashion. It looks for segments of high degrees of similarities and picks those sequences from the database that contain a segment sufficiently similar to part or all of the query sequence. The following lists the main steps of the BLAST algorithm [16]. More about the BLAST theory and algorithm can be found in [1]:

- High-scoring Segment Pairs (HSP): Given two sequences for comparison, BLAST first seeks equal length sequence segments within each, that when aligned to one another without gaps have a maximal aggregate score that cannot be increased by extension or trimming. Such locally optimal alignments are called "high-scoring segment pairs" or HSPs.
- Gapped extensions of HSPs: For any HSP that exceeds a moderate score S_g , further extend the pairing allowing gaps to be introduced. S_g is chosen so that no more than roughly one extension is invoked per 50 database sequences. Whenever a gap is opened or extended, a penalty is imposed. Find the alignment, either gapped or ungapped, with the maximal score.
- Assess statistical significance of the maximal alignment score: If it is deemed significant, the database sequence will be chosen and listed in the output.

The significance of each alignment is computed as a P value, which represents the probability of an alignment occurring with the score in question, or as an E value, which stands for the expectation value. The E value has more practical significance and is used more frequently than the P value by scientists and by other sequence analysis programs. The E value is calculated using the following formula:

$$E = Kmn e^{-\lambda S}$$

We call the left side E: the expected number of HSPs for the score S . From this formula we can see that for sufficiently large sequences of length m and n , the distribution of HSP scores is characterized by two parameters: K and λ . The lower the E value, the greater the degree of similarity between the query and comparison sequences.

Based on the above theory and algorithms, Bioinformaticians in NCBI have implemented a set of BLAST search programs to explore all of the available sequence databases regardless of whether the query is a protein or nucleotide sequence. These BLAST programs are available via the NCBI website (which can be easily configured for users' own databases on any computer that has the Apache Web Server installed), or as standalone executables that can be installed and run from the Unix Linux, or Solaris command line.

In the fungal genomics project, in order to repeatedly analyze thousands of EST sequences at once, we installed the blast executables, which we run locally against downloaded copies of the NCBI BLAST databases (and our own target database). In this "Batch" model, the input file is a large file with multiple query sequences in FASTA

format, and the output from the BLAST programs is a single plain text file, which includes all the blast results for all the queried sequences. These results can then be parsed using the BLAST Parser, which will be discussed in more detail in next section.

3.4 Motif search and Prosite database

If sequence alignment programs, such as BLAST, determine that a query protein sequence is too distantly related to any known protein then it is considered to be an unknown protein. Its relationship to known proteins can still be revealed by searching for occurrences in its sequence of particular clusters of residue types. A defined cluster of residue types is often referred to as a pattern, motif, signature or fingerprint. A number of academic groups have built databases for storing and reviewing protein motif information. Among them, PROSITE, from the ExPASy (Expert Protein Analysis System) of the Swiss Institute of Bioinformatics (SIB), may be the oldest and most popular one.

The PROSITE database consists of biologically significant patterns and profiles. It is designed in such a way that with appropriate computational tools it can rapidly and reliably help to determine to which known family of proteins (if any) a new sequence belongs, or which known domains it contains [10].

In the PROSITE database, a pattern is given as a regular expression. Regular expressions are made up of *terms, operators and modifiers*. **Terms** are the strings or substrings. **Operators** combine terms and expressions. For example, grouping with an expression like $([0-9]^+)$; repetition with $* + ?$; $\{\text{min,max}\}$ specifies how many times the proceeding expression may match. **Modifiers** can be used for changing the rules. A

regular expression evaluates to either true for strings it matches, or to false for the strings it does not match.

The basic principle behind using regular expressions to describe Motifs in the PROSITE database is the same as that used by any computer programming language (eg. Perl or Python), but is presented in a slightly different format. For example:

The motif:

```
[AC]-x-V-x(4)-{ED}
```

is used to describe a sequence in the form:

```
ala/cys- any- val- any- any- any- any-(any except glu or asp)
```

Here “any” means any amino acid.

Profiles are used when family members are very divergent. A profile consists of position specific amino acid weights and gap costs. These weights and costs (also referred to as scores) are used to calculate a similarity score between a profile and a sequence for any alignment, or between parts of a profile and a sequence [10]. In the PROSITE database, profiles are generated from multiple sequence alignments, using the members from the same protein family. When an alignment obtains a score higher or equal to a cut-off value defined by PROSITE, it signifies that a motif has been found.

Sequence patterns are very useful in detecting small regions of high sequence similarity, while profiles are more commonly used to find relationships between divergent sequences that have a small number of residues that are highly conserved within the same protein family or functional and structural domain.

The PROSITE database consists of the following two text files:

- **Prosite.dat:** is the database that contains all the information necessary for computer programs scanning sequences to retrieve Motif data.

- Prosite.doc: is the document file contains the metadata of each pattern.

To make use of the PROSITE protein Motif database, several computer programs have been developed by different academic groups. The easiest way to analyze a small set of sequences is to directly access the Motif web interface at ExPASy. In our EST annotation system, we wish to repeatedly query thousands of protein sequences against the PROSITE database at once. To accomplish this, we implemented ps_scan, which is a Perl program designed to scan PROSITE locally in the “Batch” model. The input file of this program is just a large file with multiple protein sequences in FASTA format, and the output can be selected from a list of different formats, and is specified by -o parameter at run-time (we use “-o scan” to obtain results that can be easily parsed). The following shows an example of what the analysis results look like:

```
>Contig58 : PS00847 MCM_1 MCM family signature.  
 167 - 175 GTCLIDEFD  
>Contig72 : PS00886 ILVD_EDD_1 Dihydroxy-acid and 6-phosphogluconate dehydratases signature  
 181 - 191 CDKnmPGvvmG  
.....
```

3.5 Protein signal peptide and SignalP

Another important feature for the annotation of the function of unknown sequences is the prediction of signal peptides and their cleavage sites.

A signal peptide is a short sequence that comprises the N-terminal part of the amino acid chain and that is cleaved off while the protein is translocated through the membrane. Signal peptides control the entry of virtually all proteins to the secretory pathway, both in eukaryotes and prokaryotes. Because of the special role that the protein signal peptide plays in directing the protein across a membrane, predicting the locations of signal

peptides and their cleavage sites within a protein sequence proves to be very useful for genomic research.

Instead of using traditional weight matrix methods, bioinformaticians at the CBS (Center for Biological Sequence Analysis), a facility located at the Technical University of Denmark, have developed a new method for the identification of signal peptides and their cleavage sites. The new method is based on neural networks trained with separate sets of prokaryotic and eukaryotic sequences [21].

In order to provide highly reliable predictions, they selected protein data from SWISS-PROT version 29 and divided the data into prokaryotic (further separated into Gram-positive and Gram-negative) and eukaryotic datasets. Next, redundant, homologous sequences were removed to produce final datasets that were used to train their feed-forward neural networks. (Details of data training and refining methods can be found in [21]). The trained networks provide two different scores when queried: an S-score, which can be interpreted as an estimate of the probability of the position of signal peptide; and a C-score which represents an estimation of the probability of the proposed cleavage site. (It also presents a combined score, Y-score, in the results). By a similar combination of these two pairs of artificial network results, SignalP performs significantly better than previous prediction schemas and can easily be applied to genome-wide datasets [21]. Figure 10 shows an example of SignalP's prediction results:

SignalP-NN result:

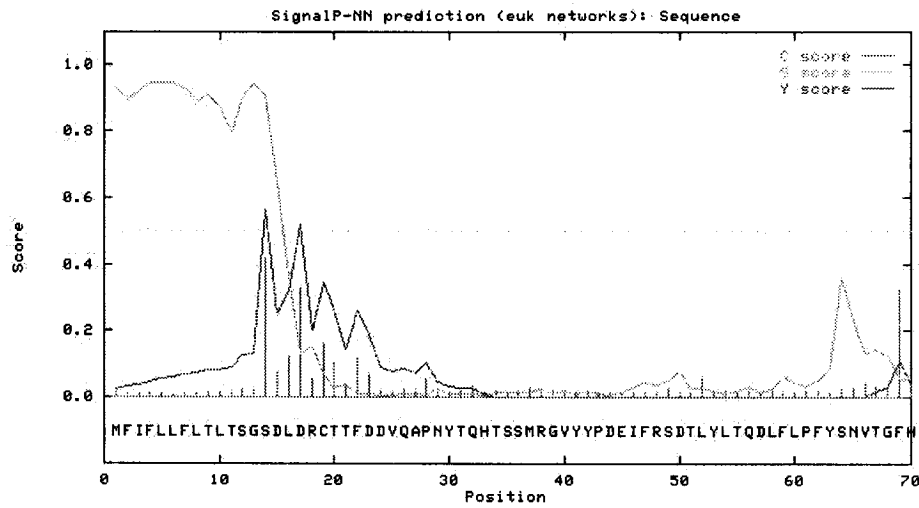


Figure 10 Example of SignalP results

In order to analyze users' own sequence data, the SignalP prediction method is available either through their provided web interface, or through a stand-alone local application, which can be obtained after signing a licensing agreement.

SignalP's web server provides a simple way to predict the possible signal peptide locations within a small number of sequences. In addition, in order to provide detailed instructions for using the submission form, it also presents a graphical plot in postscript format, like that shown in Figure 10, together with the prediction result, to help users interpret the output.

The SignalP method can also be set up locally to provide predictions for large batches of protein sequences. This was the solution pursued at the fungal genomics project, however, once we began to analyze thousands of sequences at one time several problems presented themselves. First, the current version of the SignalP program has the

limitation of being unable to analyze more than five hundred protein sequences at once. Second, to prevent the output being littered with a large number of false positive predictions, each of the query sequences must contain the N-terminal section, roughly 50-70 amino acids of the sequence, while our EST data are comprised of random sequence fragments. Therefore, with these two problems in mind, the challenge was to design a high throughput method to automatically query sequences obtained from the thousands of ESTs, in order to complement our annotation of the entire dataset with these prediction results.

To solve the first problem, we implemented a Perl script called `runSignalP.pl` to handle the large sequence files. Taking a FASTA sequence file as input, it checked the total number of sequences in the file, and if there were more than five hundred it automatically writes them to multiple files. After that, it used the Perl system call function to trigger the running of the SignalP program on each file. As its last step, it combined all the SignalP results to form a single report, which served as the input for the SignalP parser.

The solution for the second problem was a little more involved, for the following two reasons: First, because of the different status of the EST sequences, the algorithm for the selection of the N-terminal section of the query sequence needed to be different. Secondly, we needed to find a balance between missing as few hits as possible and limiting the number of false positive values. For example, for most of the sequences, searching using 50aa N-terminal sequences means we need find the longest peptide without an internal stop codon, then search using the first 50aa sequence from the first 'M' rather than using the whole sequence (to avoid getting too many false position

predictions). However, if this sequence is a short full-length sequence, which has the truncated signal peptide at the beginning, we still need to search from the beginning of this sequence.

To solve the above problems, after taking into account our understanding of our EST data as well as performing testing against the whole *Aspergillus niger* dataset, we presented the following algorithm and implemented a Perl script called `InitSignalPData.pl` to prepare the query sequences for each species before running the SignalP program:

First, the translated amino acid sequences are classified into four categories:

- A: Full-length sequences
- B: Short full-length sequences
- C: No hit sequences (after BLASTX and BLASTN)
- D: Other sequences (partial, ambiguous)

Second, the sequence data is retrieved either from the annotation database or directly from the Annotator program results (for no hit sequences), and

- For each full-length sequence, find the longest peptide without an internal stop codon, pick up sequences from the first 'M' of 50aa or to the end (if length <50 and ORF>30) from this longest peptide.
- For each short full-length sequence, select first 50aa sequences and 50aa sequences from the first 'M'.
- For each no hit sequence, if it has 50aa sequences from the first 'M', use it immediately as the query sequence; otherwise, if it does not have 50aa sequences from the first 'M', use the first 50aa from the beginning of the translation as the

query sequence. For no hit sequences, the ORFs were predicted using an in-house tool called OrfPredictor implemented by Dr. Jack Min.

- For all other sequences, do not search for a signal peptide.

3.6 Gene Ontology Mapping

Another useful resource available for the annotation of EST data is the Gene Ontology mapping. Ontology is the formalized specification of knowledge in a certain subject. Gene Ontology, initially defined by the Gene Ontology (GO) Consortium, is a dynamic, controlled vocabulary that can be applied to all organisms, despite the fact that details regarding gene and protein roles in cells are constantly being added or revised [5]. The GO Consortium has developed three separate ontologies: molecular function, biological process and cellular component. Their GO terms help to describe gene products in a standardized way and facilitate the identification of relationships and common features between genes from different species.

Using Gene Ontology to annotate proteins, more specifically, employing GO mappings to define the relationships between GO nodes and proteins, produces results that may serve as very useful references for further analysis of these proteins. Compiling GO mappings of protein data, on a large scale, can be a very frustrating task if there is no link defined between the major public protein databases and GO itself. Fortunately, to fully make use of GO resources, the GO collaborating databases, which include most of the major protein data resources, annotate their own gene products (or genes) with GO terms providing references and indicating what kind of evidence is available to support the annotations. For example, the Gene Ontology Annotation (GOA) project, run by the

European Bioinformatics Institute (EBI), is producing the mappings between all proteins in SWISS-PROT, TrEMBL and InterPro to GO, using both electronic and manual methods. After the annotation is complete, a series of index files that contain keywords to GO mappings is generated. For example:

- spkw2go is the SWISS-PROT keyword to GO mappings
- ec2go is the Enzyme Commission numbers to GO mappings
- interpro2go lists the InterPro entries and their corresponding GO terms

In order to associate our own data with GO terms, for annotation purposes, we first ran a BLASTX search against the Swiss-Prot and TrEMBL protein databases using our unigene set for each species. The BLASTX output was then parsed using an in-house parser implemented by another member in our group, Chris Beck, to build a file in GOA annotation format for connecting to a populated AmiGo database. The GOA file was parsed and loaded into the AmiGo database by scripts provided in the AmiGO distribution. The final data in the database can then be viewed and linked to from our EST annotation pages.

4. System design and implementation

4.1 System modeling and application architecture

After conducting the user survey and performing a task analysis in order to understand the research problems in this system, the most important step before system implementation is system design. There are many strategies and techniques that can be used for computer application system design, however; today, one of the most commonly used approaches to analyzing and designing information system is based on system modeling.

4.1.1 EST annotation system data modeling and data flow

The system model is a picture of a system that represents reality or desired reality. System models facilitate improved communication among users, system designers and system builders. The system modeling method is most effective for systems for which the requirements are well understood [27]. In our EST annotation system, because all the users' requirements were focused on how to obtain useful annotation data for EST sequences, it was classified as a data-centered, but process-sensitive system. To fully understand the logic and business roles of the data involved in this system, we performed both data modeling and process modeling to describe the whole system in the system design stage.

The data modeling method emphasizes the relations among knowledge building blocks, especially data. It uses diagrams to capture business roles of data and translates

them into database designs. The process modeling method focuses on describing the system internal process chain. Data flow is one of typical process model used by system designers to illustrate the flow of data through a series of system processes. The main purpose of using these two modeling techniques in our design was to answer the first critical question our EST annotation system addresses, which was: “What is the most useful information contained within these EST sequences, and how do we get it?”

4.1.1.1 Data modeling

Usually, data modeling includes two phases: First, logical modeling shows what the system does or needs to do, and excludes any technical details about the implementation. The primary tool of this stage is an ER diagram. Second, physical modeling describes how the logical model will be implemented in a given system using database schemas. Please refer to Figure 11:

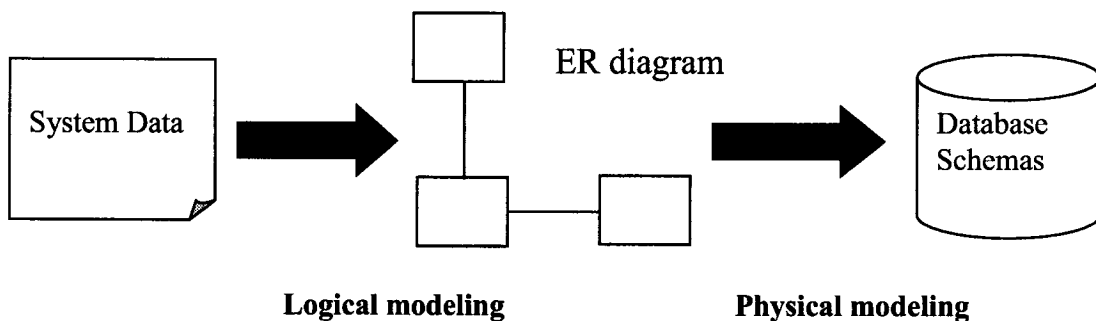


Figure 11 Data modeling stages

ER modeling produces a conceptual data model that views the real world as entities and associations between entities. A basic component of the model is the Entity-Relationship diagram, which is used to visually represent data objects.

Since the main purpose of the EST annotation system was to collect the information derived from different resources to annotate EST sequences, the entities defined in our ER model represented abstract annotation information rather than the real objects used in the biological labs; for example, the sequence full-length prediction results, sequence translations, motifs, signal peptide prediction results, and so on. The relationships defined among these entities roughly described the relations and the sequences for getting these annotation data. Figure 12 is the ER diagram that models the most important data in our system.

Each of these entities was analogous to a database table at the physical database schema implementation stage. Here, two important features should be mentioned about the actual schema design.

First, the annotation information was divided and stored according to species. In other words, each species had a separate series of annotation tables to store data, for these two reasons:

- The actual sequence data was collected, automatically analyzed, loaded, and finally published independently according to species, so we also wished to store the corresponding information separately.
- This design provided some advantages in terms of data maintenance, query and mapping implementation. Furthermore, it could facilitate future data integration.

Second, for each species, we had database schemas for both EST and unigene (contigs or singletons) annotations. The two datasets shared many common features in terms of sequence annotation. However, the data source, retrieving processes and biological contents of these two datasets had a lot of differences and cannot be mixed up in our EST annotation system.

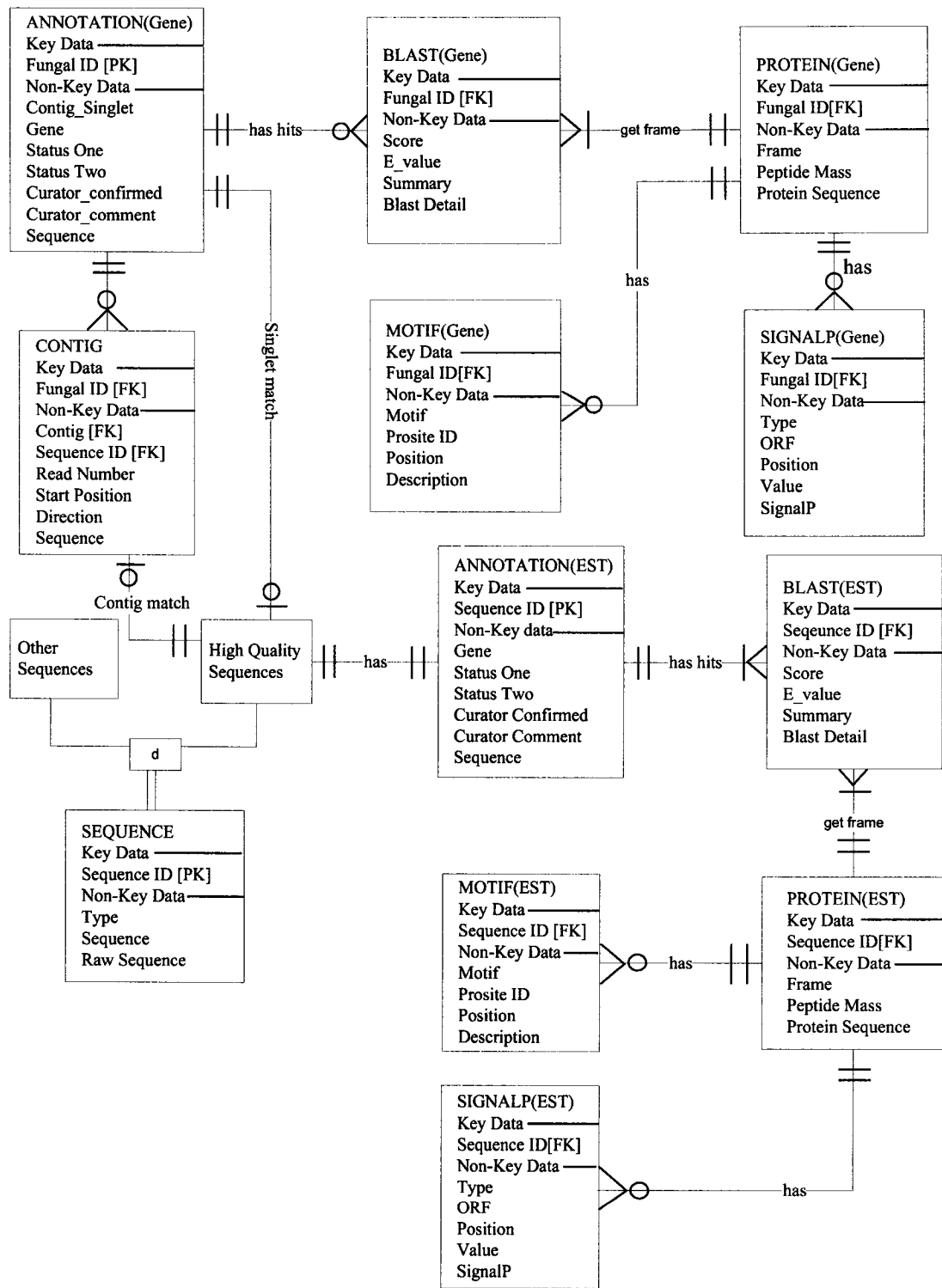


Figure 12 System ER diagram

In the above diagram, the Gene Annotation table recorded the basic annotation information about each of the unigenes, including sequence full-length prediction result, 3' coding region status, putative functional annotation, and trimmed high quality DNA sequence (for contig, it stored the consensus sequence). The Contig table was used to store the sequence alignment information for each of the contigs, including the clone name, read number, direction, start position and DNA sequence for each clone. NCBI BLAST results for each gene were recorded into the Gene BLAST table. If a unigene had a BLAST hit (with threshold E-value < 1e-5) in the Gene BLAST table, it would produce the translation information in the Gene Protein table. Then all translated sequences were to be searched by ps_scan, signalp programs to find possible motifs or signal peptides, and results were stored into the Gene Motif and SignalP tables.

Since we wanted to track all the sequences generated from each cDNA library, we used the Sequence table to record information for all of the raw sequences. In the Sequence table, a field called "Type" was used to identify the type of each sequence. A sequence may belong to one of the following four types: high quality EST, low quality sequence, short sequence and vector contaminant. For the whole high quality EST dataset, we designed another set of tables, similar to what we used for the unigene set, to store the annotation related information, so users can easily get the annotation information for both datasets through our annotation system. The appendix of this thesis lists the detailed descriptions of each database table and the major fields within these tables in the implemented system.

The database schemas in the appendix describe all the common entities we have modeled for each fungal species in the current annotation system; however, each species

may also have special table(s) to store particular information for its own purpose. For example, for *Phanerochaete chrysosporium* and *Coprinus cinereus* species, which genome sequences were available on public websites, we designed an ESTtoGenome table to store the details about how our ESTs were aligned to a known genome sequence, including sequence id, mapped scaffold number, position of alignment and detail alignment information, so users could check this information through our EST annotation system. Incidentally, since our EST annotation system is an open-ended data collection pipeline, our design allows for the addition of more tables (or entities in ER modeling), and they can be easily added to our current system as long as the nature of the data and the intended retrieval methods from different resources are clearly defined.

4.1.1.2 Process modeling

Process modeling is another important aspect of system design, which shows data flow through the system as determined by the data analysis sequence, established protocols and end-user decisions.

Since the EST annotation system is a workbench that integrates a series of sequence analysis tools for automated functional annotation, each of these tools has its own output format (or even input format). Sometimes, the output file from one application could be the input for another; in other cases it is not as simple. Therefore, a full understanding of the execution sequence and relationships among these data analysis tools serves as the foundation for building a well-working EST annotation system.

Figure 13 shows data flow for our EST annotation system. On this diagram the rounded rectangle represents a process that transforms incoming data flow into outgoing data flow, the rectangle stands for the data generated in the system, and the arrow shows the flow of data to or from a process.

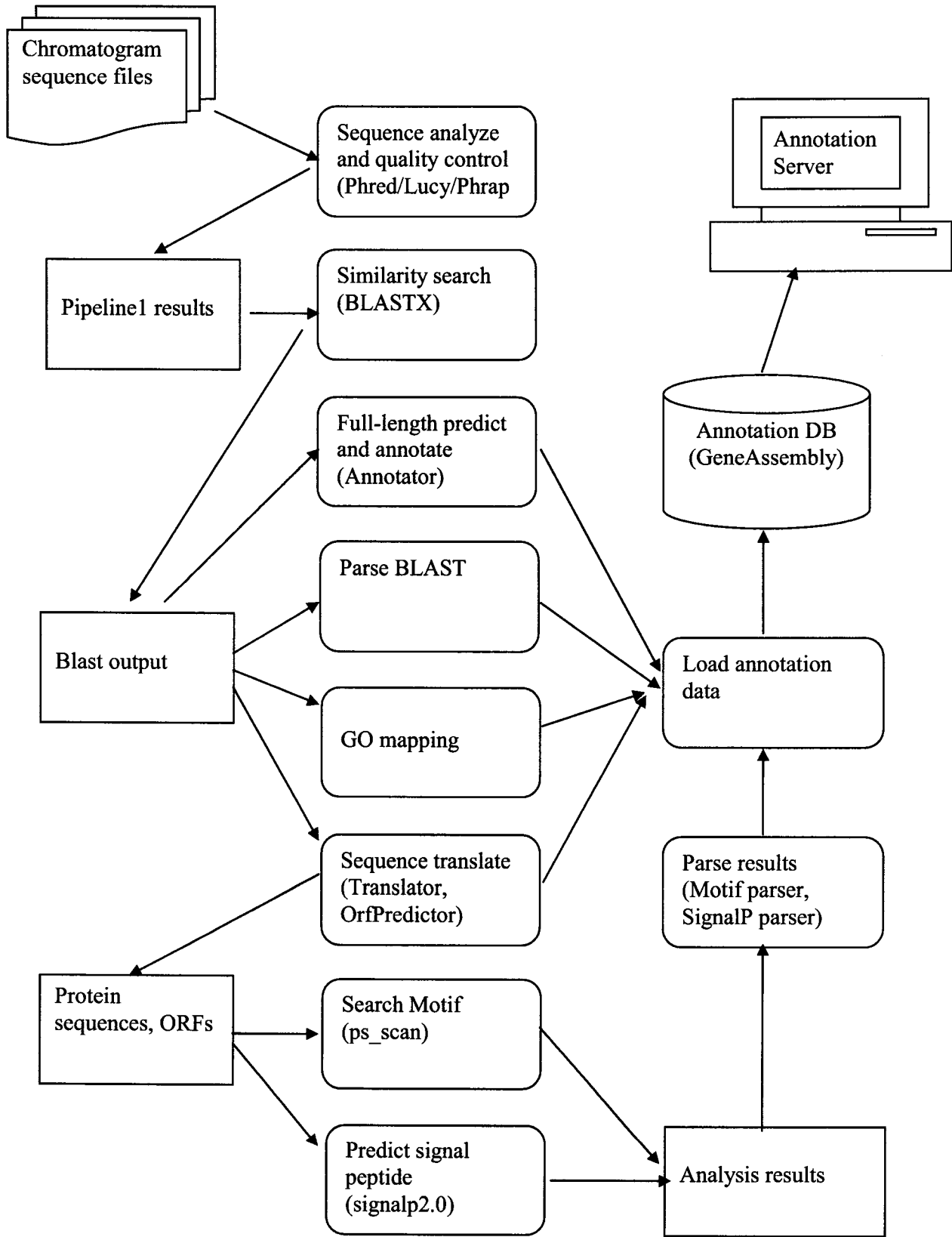


Figure 13 The data flow for annotation pipeline

4.1.2 Data presentation using web-based application

Currently, many kinds of computer systems are used to store, share, and present data in the genomics research area. For instance, the scientists in the lab are accustomed to using applications like Microsoft Word and Excel to record their data. Afterward they share the information in those files by copying them all to a centralized location on a file server. People can also easily present their data analysis results by developing static Web applications, such as HTML and JavaScript pages. However, with the unprecedented growth of data resources, it is no longer adequate to rely on this conventional file technology for organizing, storing and accessing large amounts of information on the Web. Thus, a web-based application, through which corporate databases can be linked to the Web in a manner that allows users to access data through a Web browser, are becoming a very popular approach in most of the genomics research projects.

We implemented the EST annotation system under Sun Solaris system with an Apache web server. A MySQL relational database was used to store annotation data on the server. The system used open source PHP programming language as middleware to broker and perform programmatic tasks between the web sever and the database. In the back end, Perl scripts, third-party bioinformatics programs and in-house tools were integrated to automatically retrieve information, parse the outputs and load results into the annotation database.

4.2 System construction and implementation

Since our EST annotation system is an integrated system for large-scale EST sequence analysis and annotation, which uses a variety of search and analysis tools to perform sequence analysis, the annotation data needs to be updated from time to time (normally, from 20 to 40 new DNA plates for each fungal species, each update). Due to the fact that the users of this system are biological researchers rather than computer experts, a user-friendly web-server needs to be built to allow the end-users to easily access the annotation data. For the above reasons, the next section will discuss the system implementation in terms of parser development, system automation and web-server construction.

4.2.1 Parser development

Parsers in this system consist of a series of Perl scripts, which are used to fulfill the following two purposes:

- General data parsers, which retrieve data from different resources and put useful information together for easy loading into database tables
- Special parsers, which are used to parse the analysis results from different tools (such as BLAST, SignalP), which all have different formats, and generate tab delimited text files for automatic data loading into related annotation tables

The general-purpose parsers are relatively easy to implement, as long as we know the dataflow, where to get the data and the format of the input file (for example, what a special segment mark in Phrap .ace file means). Examples of these parsers include:

- `FastatoText.pl`: a general tool for converting a file from FASTA format to text format.
- `InitAnnotation.pl`: a Perl script that reads the Annotator results, unigene file and species information to automatically initiate the annotation table and assign FID (Fungal identity number), basic annotation, sequence for each unigene.
- `InitSequence.pl`: a Perl script to read sequence information from a `lucy.seq` file, a raw sequence file, and a `lucy.seq.singlets` file and create a tab delimited text file that contains the sequence id, type (contig or singleton), sequence and raw sequence for all of the ESTs in the sequence table.
- `InitContig.pl`: a Perl script to parse a `lucy.seq` and `lucy.seq.contig` file to retrieve the contig name, read sequence name, read number, start position and sequence for each individual read within the Contig table.

The second kind of parser is used to parse data from different analysis program outputs according to the final data presentation requirement, retrieve the useful information and convert them into text files for automatic data loading in the next step. In the current system, we have the following parsers: BLAST parser, Motif parser and SignaP parser.

Prior to the implementation of a parser for a specific analysis tool, the developer should have a clear idea of its input file from two aspects: The first is to fully understand the contents of the program's output file in detail; the second is to find out the most useful information, which the end-users are interested in.

The following section uses the BLAST parser as an example, in order to illustrate how we designed and developed these parsers in our system.

NCBI BLAST is a very useful tool for performing a sequence homology search and assigning putative functions to unknown sequences. However, the output from BLAST is voluminous and in general contains much more information than is required. Thus the design and development of a proper BLAST parser has become an essential task for most bioinformatics projects.

During our research, people have developed two kinds of BLAST parsers for use by different projects.

The first kind of BLAST parser is a parser for general purposes. It reads the results file from a BLAST output, and breaks it down into a tab-delimited file. The output from this parser can include a lot of detail for both query sequence and subject sequence such as: Score, E-value, Start subject, End subject, Length, Identities, Positives, Gap. The final results can be displayed in an easy-to-read format using html tables. One obvious disadvantage of these parser programs is that the output result is not so straightforward, it includes too much information for users to make judgments themselves.

The second kind of BLAST parser is a specially designed parser, which is used to retrieve only information of interest to the users from a BLAST output. For example, in our annotation system, the annotator program has already performed efficient data mining from BLAST results for full-length prediction and putative functional annotation. The presentation of BLAST data on the annotation page was done simply because the end users sometimes liked to review it, in order to obtain a general idea of the details of the sequence similarity results as well as the sequence alignment details. To fulfill those

tasks, in our BLAST parser, we simply read the top 10 hits (if available) from the summary part and split the whole BLAST detail for each individual sequence into a plain text file as it was. This was a user-friendly design idea; when users accessed our annotation pages, by following the BLAST detail link they could easily check the alignment details to verify the data whenever they want. Otherwise, they would be forced to copy and paste their sequence from one web page to another and then blast it against the NCBI BLAST server or our local BLAST server. Figure 14 shows the BLAST results presented in our annotation pages.

Similarity search results

<BLAST detail>

Sequences producing significant alignments:	Score	E_Value
gi 224027 prf _1008149A glucoamylase G1	123	2e-27
gi 113791 sp P04064 AMYG_ASPNG Glucoamylase G1 and G2 precursor ...	123	2e-27
gi 1633184 pdb 1KUM Glucoamylase, Granular Starch-Binding Doma...	123	2e-27
gi 30025851 gb AAP04499.1 glucoamylase [Aspergillus niger]	123	2e-27
gi 40313280 dbj BAD06004.1 glucoamylase [Aspergillus awamori]	120	8e-27

Figure 14 BLAST results on annotation page

The following are the common strategies we used for developing all the parsers in our system:

- Fully understand the parameters for running applications and the detailed information in the output files.
- Focus on the users' requirement to retrieve useful data
- Output data in a proper format, which helps in presenting data in use-friendly style.

4.2.2 Pipeline automation

In the fungal genomics project, to cope with the very large volume of data, a fast and effective way of generating annotation data, via system automation, is necessary for the following two reasons:

- First, for each species, the EST annotation information needs to be updated after the analysis of every 20-40 DNA plates, so scientists can access the most up-to-date EST annotation data.
- Second, when updating the annotation information, the smaller the amount of human interaction, the smaller the potential for errors in the final results.

According to the desired extent of human interaction, the system automation can be performed on two stages: Semi-automation and Full-automation.

In this stage, we have implemented a semi-automated series of processes for deriving the annotation data, which included the following steps:

- After running the sequence assembling and quality control pipeline, manually copied the files that comprise the pipeline results and that were needed as input for different parsers, which initialized the annotation processes on a local development machine.
- Set up some analysis programs, which had special requirements for computing resources, such as, the BLAST program, which we now run on CBR (Canadian Bioinformatics Resource) PC-cluster machines, and retrieved the results after the process is done.

- Run a wrapper file, which included defining name convention for both directories and executable files, using system calls to run different analysis tools, changing input and output file formats to integrate different types together, and finally running SQL scripts in this wrapper file to automatically load data into the database tables.

In the next stage, we plan to implement a full-automation analysis pipeline to get the updated EST annotation each time new sequence data are acquired, which, from my point of view, could involve the following tasks: a) Building a centralized computer environment with enough computing power to run different analysis tools together to make the automation more easy and less error-prone. b) Synchronizing different systems and databases (both in-house and outside) to always get the most up-to-date annotation information. c) Designing and employing more powerful and flexible error checking strategies to prevent human errors in each of the data analysis steps.

Compared with the semi-automated processes, the full-automation analysis pipeline provides the following additional features:

- First, the sequence assembling and quality control pipeline can be triggered and run through a web-based user interface, and all related files needed for next step are automatically copied to the predefined directories.
- Second, the automatic downloading and reformatting (if necessary) of the most updated public databases for special programs (such as NCBI non-redundant nucleotide, or protein sequence database for BLAST, or Prosite database for Motif search) can also be triggered.

- Finally, all of the analysis tools and parsers can be defined and scheduled to run by using configuration files. The final results can then be automatically loaded into the database by running MySQL in batch mode.

4.2.3 Annotation web server design and implementation

After loading all the annotation data into a database, the last competing goal for bioinformatics was to design and implement a user-friendly web application to allow scientists to access our annotation data efficiently. The following chapter gives detailed discussion about the most important design ideas and main features presented in this system:

4.2.3.1 FID (Fungal identity number) designated to each unigene

In our annotation system, we assigned a unique FID (Fungal identity number) to each assembly sequence (either a contig sequence or a singleton sequence). The FID was composed of an acronym representing the species name, followed by a unique integer, starting with 1 and later automatically incremented as new contigs and singletons were discovered (e.g., Asp100 corresponded to *Aspergillus niger* unigene 100). The users can use this id as a key to search through our annotation database in order to retrieve all the information corresponding to this sequence (commonly referred to as a unigene). The designation of a unique FID to each assembly sequence had the following advantages:

- The naming convention for the FIDs was clear and straightforward. By quickly reviewing the list of current FIDs, the scientists can determine the current total number of unigenes that had been generated for a given species at any time.
- The FID number can serve as a foreign key when defining relationships among different annotation tables. Using the FID helped the developer to write simple and unified database queries, especially for those species that had different naming conventions for their sequence ids.
- Using unique FIDs also can simplify our life when we want to publish our data species by species.

4.2.3.2 Linking between unigene and EST annotation pages

Since the annotation information for both the EST and unigene datasets were very important references for the scientists in their work downstream, the designer of the annotation server should provide a flexible means of cross-referencing this information.

To satisfy this requirement, in our design, we provided two different pages to show the annotation for unigenes and ESTs, and the links between these pages were very clear and easy to follow:

When users search the annotation information by FID, the annotation page will present information similar to that of an EST annotation page, if the sequence belongs to a singleton. If the sequence belongs to a contig, the annotation page not only shows the annotation for the consensus sequence, but also the sequence id and sequence for each of the reads within the contig. By following the hyperlinked sequence ids, users can go

directly to the EST annotation pages to find the corresponding annotation for each read.

Figure 15 shows an example of annotation for a contig (unigene) sequence.

Aspergillus niger Annotation Information

FID number Asp1488

>FID Asp1488(Contig 1488 length:819)

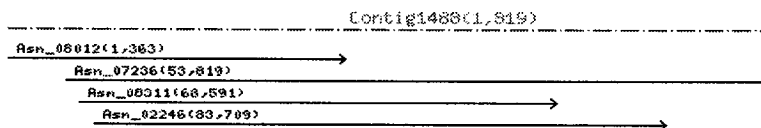
```
CATATAACAAGCTCCGCACCCCAAACTCCCCTGGCAAGTTCCTGCCCCAAATTACGTGAGACAGGGAACCATCACAGC
AACACCATGAAGAGCACCACTCTTCTTTCTTGGCCTGGGCTACCCAGTCCGCCTATTCCCTCTCTATCCACGAGCGCGA
TGAACCCGCTACTCTTCAAGTCAACTTCGAACGTCGTCAAGTCGCGGACCGGTCCCGTCCGGAAGCGATCGACGGCCTCAG
CCGACCTCGTTAACCTGGCTACGAATCTTGGCTACACGATGAACCTCACACTCGGCCTCCCGCCAGGAAGTCAGTGTG
ACGTTGGACACCGGCAGCAGCGATCTCTGGTCAATGGGGCCAACTCGTCCGTCTGCCCTGTACCGATTACGGCTCTTA
CAACTCAAGCGCTTCTTCCACCTACACCTTCTGTAACGATGAGTTTATATCCAGTATGTCGACGGCAGTGAAGCCACAG
CGGACTATGTC AACGATACTCTAAGTCTCCAATGTGACTTTGACGAACCTTCAATTTGCCGTCCGATATGACGGCGAC
TCCGAGGAGGGCGTCTCGGTATCGGATACGCCAGCAATGAAGCCAGCCAGGCCACCGTCCGTGGTGGTGAATACACAA
CTTCCCGAAGCCCTCGTCGATCAAGGCGCGATCAACTGGCCGGCTACAGTCTATGGCTGGTGAACCTCGACGAAGGAA
AAGGCACATTTTGTTCGGCGGAGTCAACACCCCAAGTACTACGGCAGCCTGCAGACCTGCCTATCGTCTCCATCGAA
GACATGTACGTTGAGTTC
```

Name	Number of Reads	Full-length prediction	Strand and 3' coding region status
Contig1488	4	Full-length	sense partial

> Contig1488 translated amino acid sequence (frame +3)

```
YVKLRTPKLPWQUPAPKLRRETGMHSMTHKSTLLSLAWATQSAVSLSIHERDEPATLQFMFERRQIADRSRRKRSTASA
DLVMLATMLGYTMMLTLGTPGQEVUSUTLDTGSSDLWVNGANSSUCPC TDYGSYNSASSTYTFUNDEFYIQYVDGSEATG
DYVNDTLKFSWTLTFQFAVAYDGDSEEGVLGIGYASNEASQATUGGGEYTFPEALUDQGAINTQPAVYSLWLDLDEGK
GTILFGGUMTAKYYGSLQTLPIUSIEDMYVEF
```

Clones	Sequence	Raw Sequence
Asn_08012	Sequence	Sequence
Asn_07236	Sequence	Sequence
Asn_08311	Sequence	Sequence
Asn_02246	Sequence	Sequence



Tentative Annotation:

Similar to aspartic protease [*aspergillus oryzae*]


Similarity search results

<BLAST detail>

Sequences producing significant alignments:	Score	E_Value
gi 21392382 dbj BAC00846.1 aspartic protease [<i>Aspergillus oryzae</i>]	218	9e-55
gi 21392384 dbj BAC00849.1 secreted aspartic protease [<i>Aspergil...</i>]	203	3e-51
gi 40738637 gb EAA57827.1 hypothetical protein AM5487.2 [<i>Asperg...</i>]	202	7e-51
gi 40549385 gb AAR87747.1 aspartic protease precursor [<i>Botryoti...</i>]	192	7e-48
gi 46122187 ref XP_385647.1 hypothetical protein FG05471.1 [<i>Gib...</i>]	188	1e-46

Number of motifs found : 1

PROSITE PATTERN

Found Motif	Position	PROSITE	Description
ASP_PROTEASE	105 - 117	 PS00141	Eukaryotic and viral aspartyl proteases active site.

SignalP results for protein signal peptide:

Query sequence:
(MKSTLLSLAWATQSAVSLSIHERDEPATLQFMFERRQIADRSRRKRSTA)

SignalP-MW result:

# Measure	Position	Value	Cutoff	signal peptide?
max. C	19	0.254	0.33	NO
max. Y	19	0.350	0.32	YES
max. S	1	0.818	0.82	NO
mean S	1-18	0.595	0.47	YES

Most likely cleavage site between pos. 18 and 19: AVS-LS

Figure 15 An example of a unigene annotation page

In contrast, if users search for each individual EST by sequence id, the FID can easily be determined as each EST should correspond either to a singleton or be one of several reads that comprises a contig. When users click on the hyperlinked FID listed in the EST's annotation, the server will automatically retrieve the annotation page for the unigene. Hence, the users can easily get the annotation information for all the sequences that have been aligned with a given sequence to form a unigene.

4.2.3.3 Search engine design

When users enter the EST annotation system, the first page provides a summary table of the most up-to-date annotation data for the given species (as the annotation is compiled separately by species), such as, current EST and unigene totals, the number of known genes referenced in BLASTX and BLASTN results, and the total number of genes encoded with a signal peptide. Under this summary table, the system provides a series of search engines enabling the users to easily retrieve useful information from our annotation database.

The search engine design was the most crucial part of our annotation server implementation. A set of well-designed search options facilitated searching the EST annotation database from different aspects. Based on our understanding of the users' requirements, we provided three groups of search engines in the current system. In total there were five search options provided to the users for accessing the annotation data (see Figure 16), and they were:

- Search by id: includes options to search the annotation database by FID and by Sequence Id.
- Search by key word: includes options to search by Gene name and search by Category.
- Additional search: includes options to search by EC number and GO id.

Phanerochaete chrysosporium annotation information

<Known Genes SignalP> <No hit Genes SignalP>

Search sequence information by FID number:
(FID is a unique id assign to each assembly sequence)

Current FID number is (Pchr1 to Pchr4801)

Search by sequence ID:
(Using sequence id like PchrSEQ1009)

Search by key words:
(Key words search can be performed by use gene name such as: ribonucleotide reductase, glucosidase or by Category such as: lip)

Search by GO id, such as GO:0003925 :

GO:

Search by EC number, such as EC 6.1.1.17 :

Figure 16 Search engines in annotation summary page

By using the first set of search options, the users can find all the annotations, including: the nucleotide sequence, full-length prediction, translation, putative function assignment, motif, signal peptide prediction and so on, for either individual clones or the assembly sequences that they are interested in. As mentioned in the previous chapter, the

system also cross-references these two kinds of annotation information, enabling users to easily flip back and forth between them.

Performing searches by key word is decidedly the most useful function in this system. Most of the time, scientists want to search through the annotation database using the name of a gene that they are interested in, in order to find clones or genes related to it. In addition to the search by gene name function, in our implementation of the system we provided an additional function to enhance the potential of the search engine, allowing users to search by category. A gene category uses a three-letter symbol to represent a group of gene names or EC numbers, all of which belong to the same gene family in our target gene list. A category-to-gene look-up table was added into the annotation database. When users search the annotation database by a category, a specifically defined database query function will help users to search through the annotation database using all the gene names and EC numbers represented by this category, and retrieve all the possible related genes or clones. In this way, users can quickly get an idea of how many genes for a specific gene family have already been clarified in the current libraries.

Two additional search functions allow users to search our annotation database either by GO id or EC number.

The GO id is the Gene Ontology number assigned to a group of Genes through a GO mapping. By searching with a GO number, people can find a number of genes within a specific species that have the same molecular function, share a common biological process or are associated with the same cellular components. A link to AmiGO is included with each GO (Gene Ontology) annotation. It displays a full DAG (direct acyclic graph) representation of the query results.

The EC number refers to the Enzyme Classification (or Commission) number. As a system of enzyme nomenclature, every EC number is associated with a recommended name for the respective enzyme. Since the method of using EC numbers to classify enzymes based on type of reaction catalyzed, substrate and reagent/co-factors are widely adopted in biological research and the EC numbers are associated with most of Swiss-Prot and TrEMBL entries. Sometimes, a search by EC number may be very useful in supporting those working in the lab when classifying enzymes.

4.2.3.4 User-friendly features

Since the final users of the EST annotation system were the scientists, who needed to access the annotation data frequently on a daily basis, the main design strategies for user interfaces of this EST annotation system focused on facilitating ease-of-use when performing queries and providing enough documentation to allow them to compile references to support their lab work. To achieve these goals, in our final implementation, the EST annotation system presented the following user-friendly features, which made it a very efficient system for the scientists:

- A User-friendly web design for sequence data presentation

As most biologists know, sequence FASTA format is a compact and simple method of storing DNA and protein sequences as text files that can be read by most sequence analysis programs. The FASTA sequence format consists of a sequence name and description on a single line starting with the greater than symbol '>', followed by the

sequence data in the second line. To make our EST annotation pages more user-friendly, all the nucleotide and protein sequences shown on the annotation pages were following this FASTA format with column widths of 80 characters. This convention had two advantages: first it made it very easy for the users to calculate the full sequence length of a displayed sequence. Second, if a scientist wished to do more analysis using an annotated sequence, they could directly copy and paste the sequence from the annotation page and use it in another application.

- Hyperlinks to the original databases allow users to get more information

Within the annotation pages, wherever applicable, cross-references from the original databases were provided as hyperlinks, giving users more references for the annotations. For example, the gi number (link to NCBI) and Prosite entries (link to PROSITE Motif database) were all cross-linked.

- Color-coding to mark the important regions of a sequence in the annotation data

To help users quickly identify the most interesting regions, such as the characterized Motifs or predicted signal peptide (and its cleavage site) in a query sequence, we used different color codes when presenting the annotation data to highlight regions within the whole sequence. Figure 17 shows an example of an identified predicted signal peptide in a protein sequence, taken from the annotation page:

> An_2230.D01 Predicted amino acid sequence (frame +2)

INLRLILSMIFVDGELSLSQMAAYAGCSKPTIIAIRSNLRMFGSVRTPPVKCRPRRLTTAMMDALCGHLLLELDLYLHE
MQVFLDELCLYVSTSTV

(Query sequence: INLRLILSMIFVDGELSLSQMAAYAGCSKPTIIAIRSNLRMFGSVRTPPV)

SignalP-NN result:

# Measure	Position	Value	Cutoff	signal peptide?
max. C	20	0.243	0.33	NO
max. Y	20	0.381	0.32	YES
max. S	12	0.876	0.82	YES
mean S	1-19	0.721	0.47	YES

Most likely cleavage site between pos. 19 and 20: SLS-QM

***Red color showing the query sequence used for searching of signal peptide

***Green color showing the signal peptide within the query sequence

Figure 17 Predicted signal peptide in a protein sequence

- Providing a graphic representing the sequence alignment information

Although plain text and HTML tables are sufficient for presenting the contents of the annotation in most cases, sometimes, a graphical view of the information is more efficient. For example, for each unigene sequence built by the PHRAP assembly program, we annotated the unigene's consensus sequence, and listed all the clones (and their nucleotide sequences), which form the contig. In addition to this information, the users still wanted to be provided with a graphical representation of how these clones were aligned together. They wanted to know the start position of each clone (relative to the beginning of the consensus sequence), as well as its length and direction. To fulfill this requirement, we took advantage of the PHP GD library to dynamically generate sequence alignment graphs from data in the annotation database. A specially designed PHP class was used to read through the Contig table, automatically draw the alignment picture on screen to show the consensus sequence, and display all the reads in their appropriate

positions within the alignment. Figure 18 provides an example of a sequence alignment graph taken from our annotation:

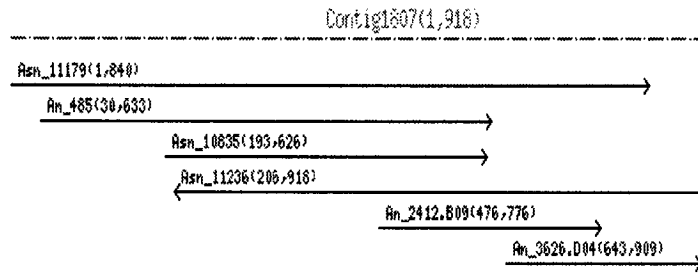


Figure 18 Sequences aligned to form a contig

Right now, scientists in Concordia fungal genomics project are working on this system every day, they are willing to provide the feedback about their feeling and evaluations of the whole system. This helps us to build an always-evolving annotation system to include more useful and user-friendly features to support further genomic research.

5. Conclusion and possible future extensions

In conclusion, when compared with the public sequence annotation systems we have studied, our EST annotation system includes not only the common features for EST data storage, function assignment and visualization, but also some important features that are not found in other EST annotation systems, such as, EST sequence full-length prediction, protein signal peptide prediction. In term of building a user-friendly annotation system, our EST annotation presents the following interesting features:

- First, the unigene-based and sequence-based data integration allows users to check the data from two different points of view, both of which provide very useful references for further genomics research.
- Second, parsers are well developed to derive only the most interesting information from different sequence analysis results.
- Third, well-designed search engines and annotation pages allow users to easily access, retrieve and use all kind of annotation information and cross-references that they are interested in.

Currently, all the features and functions mentioned above have been fully implemented. The annotation system is in use and is fulfilling its role in supporting the biological research in the fungal genomics project. However, as an open-ended data collection pipeline, additional useful features are being evaluated as possible extensions to be included in the next version of the system. The following lists some extensions that might be incorporated into the system in the near future.

5.1 Annotation of protein domains using InterProScan

Using a sequence similarity search with a proper threshold (for example, E-value < 1e-5 for BLASTX against nr database), we can assign putative functionality to most (usually between 75 and 90%) of the ESTs in our database. This leaves us with the remaining 10 to 25 percent of the genes from each fungal species being assigned no known function. For example, in the final *Aspergillus niger* dataset, we have a total of 912 uncharacterized genes amongst the 5202 unigenes. In order to identify distant relationships in these novel sequences, the protein signature databases are very helpful.

In recent years, several protein signature databases and recognition methods have been built by different bioinformatics research projects. Although all of these resources share the common interest in protein sequence identification, each of them has its own strengths and weaknesses while addressing the problem from different aspects.

InterPro is developed to integrate these data resources for protein families, domains and functional sites, and provide an easy and reliable way to identify protein family and functionality. In its current stage, InterPro includes the following member databases:

- PROSITE (Hofmann, K. et al. 1999)
- PRINTS (Attwood, T. K. et al. 2000)
- Pfam (Bateman, A. et al. 2000)
- ProDom (Corpet, F. et al. 1999)
- SMART (Schultz, J. et al. 2000)
- TIGRFAMs (Haft, D.H. et al. 2001)

InterPro systematically integrates the above databases and creates a unique, non-redundant characterization of a given protein family, domain or functional site. Each InterPro entry has been assigned a unique accession number and includes a functional description, annotation, literature references and links back to the relevant member databases [3].

To facilitate the searching of InterPro databases, a very scalable and extensible tool called InterProScan has been developed by EBI to scan input protein sequences against the protein signatures from the InterPro member databases. InterProScan can be accessed through EBI's web interface or it can be set up locally. The stand-alone version of InterProScan is implemented in Perl and is designed to cope with many unknown sequences at one time. Since each InterPro entry provides a detailed functional annotation for the corresponding query sequence, as well as references to matches in the parent databases, the InterProScan results for each gene could be very useful information to be added to our EST annotation system.

Currently, the bottleneck for integrating InterPro results into our annotation system is lacking of enough computing power to run it for the whole unigene set for each species. As a temporary solution, we are trying to only run it against FUN (function unknown) gene dataset, and add the results to our annotation database to make these genes more meaningful to the scientists.

5.2 Gene pathway and KEGG

Using microarray technology to study, on a large scale, gene expression, as well as the availability of genome sequences for the fungal species we are interested in (such as,

Phanerochaete chrysosporium and Coprinus), it is possible for us to monitor expression patterns across the whole set of genes in a genome. To effectively analyze a collection of gene catalogs spanning a whole genome (or a partial genome), it is necessary for us to integrate them with pathways or complexes, which represent the higher order biological functions.

KEGG (Kyoto Encyclopedia of Genes and Genomes) is a knowledge base for systematic analysis of gene functions, linking genomic information with higher order functional information [15]. The main purpose of the KEGG project is to provide a computer representation of functional data, in relation to a network of interacting molecules in the cell such as metabolic pathway, various regulatory pathway and molecular assemblies. It also provides many analysis tools, such as, pathway maps, identifying gene clusters conserved in two genomes, and pathway reconstruction.

Prior to making use of KEGG pathway databases to annotate our own expression genes, the following prerequisites should be accomplished by both biologists and bioinformaticians in this project:

- Interesting gene clusters should be generated by using microarray;
- Genome sequence annotation databases should be adopted for systematic annotation of gene functions;
- The algorithm for Mapping from our own gene clusters to KEGG pathway should be implemented.

The integration of KEGG pathway data into our sequence annotation system could be a very valuable and practical task for our bioinformatics research in the near future.

6. References

1. Altschul S.F., Gish W., Miller W., Myers E.W. & Lipman D.J. "Basic local alignment search tool", *J. Mol. Biol.* 1990; 215:403-410.
2. Andrade M.A., Brown N.P., Leroy C, Hoersch S, de Daruvar A, Reich C, Franchini A, Tamames J, Valencia A, Ouzounis C, Sander C. "Automated genome sequence analysis and annotation", *Bioinformatics.* 1999 May. 15 391-412.
3. Apweiler R, Attwood T.K., Bairoch A, Bateman A, Birney E, Biswas M, Bucher P, Cerutti L, et al. "The InterPro Database, an integrated documentation resource for protein families, domains and functional sites", *Nucleic Acids Res.* 2001 Jan 1; 29(1):37-40.
4. Ayoubi P, Jin X, Leite S, Liu X, Martajaja J, Abduraham A, Wan Q, Yan W, Misawa E, Prade R.A. "PipeOnline 2.0: automated EST processing and functional data sorting", *Nucleic Acids Res.* 2002 Nov 1; 30 (21):4761-9.
5. Camon E, Barrell D, Brooksbank C, Magrane M and Apweiler R. "The Gene Ontology Annotation (GOA) Project: Implementation of GO in SWISS-PROT, TrEMBL, and InterPro", *Genome Research* 2003 Apr; 13 (4): 662-672.
6. Chen Y.Q. "Pipeline for the quality control of sequencing", Concordia University, Master Thesis (2002).
7. Chou H.H, Holmes M.H. "DNA sequence quality trimming and vector removal", *Bioinformatics.* 2001. 17(12):1093-1104.

8. EnsEMBL core schema documentation.
http://www.ensembl.org/Docs/schema_description.html
9. Ewing B., Hillier L., Wendl M.C., Green P. “Base-calling of automated sequencer traces using phred. I. Accuracy assessment”, *Genome Res.* 1998 Mar;8(3):175-85.
10. Falquet L., Pagni M., Bucher P., Hulo N., Sigrist C.J, Hofmann K., Bairoch A. “The Prosite database, *its status in 2002*”, *Nucleic Acids Res.* 30:235-238(2002).
11. Frishman D, Albermann K, Hani J, Heumann K, Metanomski A, Zollner A, Mewes HW. “Functional and structural genomics using PEDANT”, *Bioinformatics.* 2001 Jan;17(1):44-57.
12. Green P. Documentation for Phrap and Cross-match
<http://www.genome.washington.edu/UWGC/analysistools/Phrap.cfm>
13. Hotz-Wagenblatt A, Hankeln T, Ernst P, Glatting K.H, Schmidt E.R., Suhai S. “ESTAnnotator: a tool for high throughput EST annotation”, *Nucleic Acids Res.* 2003 July 1; 31(13): 3716–3719.
14. Hubbard T., Barker D., Birney E., Cameron G., Chen Y., Clark L., Cox T., Cuff J., Curwen V., Down T. et al. “The Ensembl genome database project”, *Nucleic Acids Research.* 2002; 30 38–41.
15. Kanehisa M, Goto S. “KEGG: Kyoto Encyclopedia of Genes and Genomes”, *Nucleic Acids Res.* 2000 Jan; 28, 27-30.
16. Leung M.Y. “Basic Local Alignment Search Tool”,
<http://www.math.utep.edu/Faculty/mleung/teaching/s697s00/trans/blast.pdf>.
17. Mao C, Cushman J.C., May G.D., Weller J.W. “ESTAP – an automated system for the analysis of EST data”, *Bioinformatics* 2003, 19(13):1720-2.

18. Matukumalli L.K., Grefenstette J.J., Sonstegard T.S., Van Tassell C.P. "EST-PAGE--managing and analyzing EST data", *Bioinformatics* 2004 Jan 22; 20(2): 286-8.
19. Min X.J., Butler G., Storms R., Tsang A. "TargetFinder and Annotator: a Simple Approach for Finding Full-length Target cDNAs and for Annotating EST Sequence", Concordia University, un-published (2003).
20. National Center of Biotechnology Information A Science Primer
<http://www.ncbi.nlm.nih.gov/About/primer/>
21. Nielsen H., Brunak S., Engelbrecht J. and von Heijne G. "Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites", *Protein Eng.* 1997 Jan;10(1):1-6.
22. Petty Y. The tutorial on DNA structure, replication, transcription, and protein synthesis. <http://www.ncc.gmu.edu/dna/replicat.htm>
23. Quackenbush J., Cho J., Lee D., Liang F., Holt I., Karamycheva S., Parvizi B., Pertea G., Sultana R., White J. "The TIGR Gene Indices: analysis of gene transcript sequences in highly sampled eukaryotic species", *Nucleic Acids Research.* 2001 Jan 1;29(1):159-64.
24. Quackenbush J., Liang F., Holt I., Pertea G., Upton J. "The TIGR Gene Indices: reconstruction and representation of expressed gene sequences", *Nucleic Acids Research.* 2000 Jan 1;28(1):141-5.
25. Saccharomyces Genome Database. <http://www.yeastgenome.org/>

26. Thomas P.D., Campbell M.J., Kejariwal A, Mi H, Karlak B, Daverman R, Diemer K, Muruganujan A, Narechania A. "PANTHER: a library of protein families and subfamilies indexed by function", *Genome Res.* 2003 Sep;13(9):2129-41.
27. Whitten J.L, Bentley L.D., Dittman K. "Systems Analysis and Design Methods", Sixth Edition 2003.
28. Xu H, He L, Zhu Y, Huang W, Fang L, Tao L, Zhu Y, Cai L, Xu H, Zhang L, Yu H, Zhou Y. "EST Pipeline System: detailed and automated EST data processing and mining", *Geno Prot Bioinfo* 2003 Aug; 1(3):236-242.

7. Appendix

The following are EST annotation system database schemas and the detailed descriptions of each field. We duplicate the tables for each organism. In our design, each fungal species has 12 tables to store annotation and related information. The real table name is like Organism_TableName. The following part uses *Aspergillus niger* related tables as an example:

```
CREATE TABLE Niger_annotation (  
    FID                varchar(10) NOT NULL default '',  
    Contig_Singlet     varchar(30) NOT NULL default '',  
    Gene               text        default '',  
    StatusOne          varchar(30) NOT NULL default '',  
    StatusTwo          varchar(30) NOT NULL default '',  
    Sequence           text        NOT NULL default '',  
    CuratorConfirmed   char(1)    default '',  
    CuratorComment     text        default '',  
    Primary key (FID),  
) TYPE=MyISAM;
```

```
CREATE TABLE Niger_annotation_EST (  
    Seq_id             varchar(30) NOT NULL default '',  
    Gene               text        default '',  
    StatusOne          varchar(30) NOT NULL default '',  
    StatusTwo          varchar(30) NOT NULL default '',  
    CuratorConfirmed   char(1)    default '',  
    CuratorComment     text        default '',  
    Primary key (Seq_id)  
) TYPE=MyISAM;
```

Two annotation tables are used to store the basic annotation information for both unigene and EST sequences. They include the following fields:

FID: stands for Fungal Identity number, it is a unique id assigned to each fungal unigene. This is the primary key field in the annotation table, and serves as the foreign key in other related tables. The advantages of defining the FID in this system have been discussed in chapter 4.

Contig_Singlet: stores contig or singleton name of each unigene. If the unigene is a contig, it stores the contig name; if the unigene is a singleton, it stores its sequence name.

Gene: stores the tentative functional annotation assigned to a sequence. The information is derived from Annotator program.

StatusOne: is the sequence full-length prediction result derived from the Annotator program, which points out whether the cloned cDNA includes intact coding regions (are of full-length)[19].

StatusTwo: is the sequence 3' coding region status derived from the Annotator program, which points out whether the sequence 3' coding region is completely obtained [19].

Sequence: is used to store the nucleotide sequence. For contig, it stores the consensus sequence after assembling; for singleton, it stores the trimmed high quality sequence.

CuratorConfirmed: stores the information of whether curators have manually checked the annotation information for this unigene (or EST).

CuratorComment: stores curator's manual annotation information or comments.

Seq_id (in EST annotation table): is used to store the unique sequence id for each individual EST sequence.

```
CREATE TABLE Niger_contigs (  
  FID          varchar(10)  NOT NULL default '',  
  Contig       varchar(20)  NOT NULL default '',  
  Seq_id       varchar(30)  NOT NULL default '',  
  ReadNumber   varchar(3)   NOT NULL default '',  
  StartPosition varchar(10) NOT NULL default '',  
  Direction    varchar(2)   NOT NULL default '',  
  Sequence     text         NOT NULL default '',  
  Primary key (FID),  
  Index (Contig)  
) TYPE=MyISAM;
```

The contigs table is used to save the information for all assembled sequences (contigs). A contig could have one or multiple records in this table according to how many sequences (or reads) are aligned to form this contig. Each row in this table records the information for one aligned sequence (or read). The information is derived directly from the Phrap output .ace file. It includes the following fields:

Contig: is the contig name for each assembled sequence.

Seq_id: is the sequence id for each read to form the contig.

ReadNumber: points out how many sequences (or reads) are aligned to form a specific contig.

StartPosition: is the padded start consensus position, which points out the relative start position of the read sequence against the consensus sequences.

Direction: indicates whether the aligned sequence is a 5' end sequence or 3' end sequence.

Sequence: stores the nucleotide sequence for the aligned sequence in a contig.

```
Create table Niger_protein (
  FID          varchar(10) NOT NULL default '',
  Frame        char(2)     default '',
  EstimatedPeptideMass varchar(10) NOT NULL default '',
  ProteinSequence text      NOT NULL default '',
  Primary key (FID),
) TYPE=MyISAM;
```

```
Create table Niger_protein_EST (
  Seq_id       varchar(40) NOT NULL default '',
  Frame        char(2)     default '',
  EstimatedPeptideMass varchar(10) NOT NULL default '',
  ProteinSequence text      NOT NULL default '',
  Primary key (Seq_id)
) TYPE=MyISAM;
```

The protein tables are used to save sequence translation information for both unigene and EST sequences, which are derived directly from our in-house sequence translation program. The following are the descriptions of each field:

Frame: indicates which reading frame is used to do the translation for this DNA sequence, it could be one of the following numbers: +1, +2, +3, -1, -2 and -3.

EstimatePeptideMass: indicates the estimated protein molecule weight for each translated sequence, which is an important reference for protein analysis.

ProteinSequence: is the translated amino acid sequence for this DNA sequence.

```
CREATE TABLE Niger_BLAST (  
    FID          varchar(10) NOT NULL default '',  
    Score        double(6,1) NOT NULL default 0.0,  
    E_value      varchar(10) NOT NULL default '',  
    Summary      text        NOT NULL default '',  
    Blast        text        NOT NULL default '',  
    Primary key (FID),  
) TYPE=MyISAM;
```

```
Create table Niger_BLAST_EST (  
    Seq_id       varchar(40) NOT NULL default '',  
    Score        double(6,1) NOT NULL default 0.0,  
    E_value      varchar(10) NOT NULL default '',  
    Summary      text        NOT NULL default '',  
    Blast        text        NOT NULL default '',  
    Primary key (Seq_id),  
) TYPE=MyISAM;
```

The BLAST tables are used to store the BLAST (BLASTX and BLASTN) results for both the unigene and EST sequences. The data are obtained by running BLAST parser to parse the NCBI BLAST results. The following are the descriptions of each field:

Score: is the score of a BLAST hit in the results, which is a measurement of the statistical properties of the similarity of two compared sequences.

E-value: is the expectation value of a BLAST hit. The number of different alignments with scores equivalent to or better than S that are expected to occur in a database search by chance. The lower the E value, the more significant the score.

Summary: records the summary line retrieved from the blast results.

Blast: stores the whole blast result as it is in the plain text file, including the detailed sequence alignment information.

```
CREATE TABLE Niger_Motif (
  FID          varchar(10) NOT NULL default '',
  Motif        varchar(30) NOT NULL default '',
  Prosite_id   varchar(10) NOT NULL default '',
  Position     text        NOT NULL default '',
  Description   text        NOT NULL default '',
  Primay key (FID),
) TYPE=MyISAM;

Create table Motif_EST (
  Seq_id       varchar(40) NOT NULL default '',
  Motif        varchar(30) NOT NULL default '',
  Prosite_id   varchar(10) NOT NULL default '',
  Position     text        NOT NULL default '',
  Description   text        NOT NULL default '',
  Primay key (Seq_id),
) TYPE=MyISAM;
```

Motif tables record the possible motifs found in query sequences for both the unigene and EST datasets. The following are the descriptions of each field:

Motif: stores the name of the motif found in a protein sequence.

Prosite_id: stores the Prosite database accession number corresponding to the found motif, can be used to build a cross-reference link on the annotation page.

Position: indicates the position of the motif found within the query sequence.

Description: stores a detailed description of the protein family that is associated with the protein sequence motif.

```

CREATE TABLE Niger_SignalP (
  FID          varchar(10) NOT NULL default '',
  Type         varchar(30) NOT NULL default '',
  ORF          text        NOT NULL,
  Position     varchar(10) NOT NULL default '',
  Value        double(7,3) NOT NULL default '0.000',
  SignalP      text        NOT NULL,
  Primay key (FID),
) TYPE=MyISAM;

```

```

Create table Niger_SignalP_EST (
  Seq_id       varchar(40) NOT NULL default '',
  Type         varchar(30) NOT NULL default '',
  ORF          text        NOT NULL,
  Position     varchar(10) NOT NULL default '',
  Value        double(7,3) NOT NULL default '0.000',
  SignalP      text        NOT NULL,
  Primay key (Seq_id)
) TYPE=MyISAM;

```

SignalP tables are used to store protein signal peptide(s) found by the SignalP program for a given protein sequence in both the unigene and EST datasets. The following are the descriptions of each field:

Type: records the type of the query sequence, which could be “Known Gene” or “No Hit” sequence according to whether it has a hit (or hits) after running BLASTX against NCBI nr database.

ORF: is the open reading frame used for searching of protein signal peptide(s).

Position: shows the position of the predicted signal peptide within a protein sequence.

Value: is the mean value report by SignalP program based on neural networks. For a query sequence, if this value is more than the cut-off (>0.47), it signifies a possible protein signal peptide, as predicted by the SignalP program.

SignalP: contains the detailed SignalP prediction result for the query sequence.

```

CREATE TABLE Niger_sequence (
  Seq_id       varchar(30) NOT NULL default '',
  Type         varchar(20) NOT NULL default '',

```

```
Sequence          text          NOT NULL default '',
RawSequence       text          NOT NULL default '',
Primary KEY (Seq_id)
) TYPE=MyISAM;
```

The sequence table is used to store the information for all the raw sequence data that passed through our sequence assembling and quality control pipeline, including those sequences that do not belong to the high quality sequences (ESTs). This allows scientists to easily check each individual sequence. It includes the following fields:

Seq_id: is a unique id for each individual sequence.

Type: stores the category of each sequence, which could be contig, singleton, low quality sequence, short sequence or vector contaminant sequence.

Sequence: for ESTs (which type is contig or singleton), the field is used to store the trimmed high quality sequence; for sequences that belong to other categories, it contains the untrimmed sequence.

RawSequence: stores the untrimmed DNA sequences.

8. Glossary

Annotation: Sequence annotation is the process of adding biological information to a genome sequence. This is a very complex task, and the process for accomplishing it is rapidly evolving.

BLAST (Basic Local Alignment Search Tool): A sequence comparison algorithm that is optimized for speed and used to search sequence databases for optimal local alignments to a query.

cDNA (complementary DNA): A DNA sequence obtained by reverse transcription of a messenger RNA (mRNA) sequence.

cDNA library: A collection of DNA sequences generated from mRNA sequences. This type of library contains only protein-coding DNA (genes) and does not include any non-coding DNA.

Codon: Sequence of three nucleotides in DNA or mRNA that specifies a particular amino acid during protein synthesis; also called a triplet. Of the 64 possible codons, 3 are stop codons, which do not specify amino acids.

Consensus sequence: A derived nucleotide sequence that represents a family of similar sequences. Each base in the consensus sequence corresponds to the base most frequently occurring at that position, in the real sequences.

Contig: A contiguous segment of the genome made by joining overlapping clones or sequences. A clone contig consists of a group of cloned (copied) pieces of DNA representing overlapping regions of a particular chromosome. A sequence contig is an extended sequence created by merging primary sequences that overlap. A contig map shows the regions of a chromosome where contiguous DNA segments overlap. Contig maps provide the ability to study a complete and often large segment of the genome by examining a series of overlapping clones, which then provide an unbroken succession of information about that region.

DNA: A double-stranded molecule that encodes genetic information for any living organism.

DNA Sequencing: The experimental process of determining the nucleotide sequence of a region of DNA.

EC number: A number assigned to a type of enzyme according to a scheme of standardized enzyme nomenclature developed by the Enzyme Commission of the Nomenclature Committee of the International Union of Biochemistry and Molecular

Biology (IUBMB). EC numbers may be found in ENZYME, the Enzyme nomenclature database, maintained at the ExpASy molecular biology server.

E-value (Expectation value): The number of different alignments with scores equivalent to or better than S that are expected to occur in a database search by chance. The lower the E value, the more significant the score.

EST (Expressed Sequence Tag): ESTs are short (usually approximately 300–500 base pairs), single-pass sequence reads from cDNA. Typically, they are produced in large batches. They represent the genes expressed in a given tissue and/or at a given developmental stage. They are tags (some coding, others not) of expression for a given cDNA library. They are useful in identifying full-length genes and in mapping.

Exons: The protein-coding DNA sequences of a gene.

FASTA format: A sequence in FASTA format begins with a single-line description, followed by lines of sequence data. The description line is distinguished from the sequence data by a greater-than (">") symbol in the first column. It is recommended that all lines of text be shorter than 80 characters in length.

Full-length sequence: A sequence is considered full-length when it satisfies one of the following two criteria. (1) The sequence has a 5' stop codon followed by a start codon (2)

The sequence does not have a 5' stop codon but there is an in-frame start codon present prior to codon 10 of the aligned subject sequence.

Introns: The DNA base sequences interrupting the protein- coding sequences of a gene; these sequences are transcribed into RNA but are cut out of the message before it is translated into protein.

Motif: A conserved element of a protein sequence alignment that usually correlates with a particular function. Motifs are generated from a local multiple protein sequence alignment corresponding to a region whose function or structure is known. It is sufficient that it is conserved, and is hence likely to be predictive of any subsequent occurrence of such a structural/functional region in any other novel protein sequence.

Sequence alignment: the process of lining up two or more sequences to achieve maximal levels of identity (and conservation, in the case of amino acid sequences) for the purpose of assessing the degree of similarity and the possibility of homology.

ORF (Open Reading Frame): A sequence of DNA following an initiation codon that does not contain a stop codon. Detection of an open reading frame in DNA implies the presence of a gene that codes for a protein.

Signal peptide: are recognized by a signal recognition particle that draws the ribosome to the membrane surface by interaction with a docking protein.

Vector: A cloning vector that is engineered to allow the expression of protein from cDNA. The expression vector provides an appropriate promoter and restriction sites that allow insertion of cDNA.