Small Genome Annotation and Data Management at TIGR

Michelle Gwinn, William Nelson, Robert Dodson, Steven Salzberg, Owen White

Abstract
TIGR has developed, and continues to refine, a comprehensive, efficient system for small genome annotation. The Glimmer gene finding software identifies open reading frames most likely to code for genes. The protein sequences from these genes are searched against a nonredundant protein database, protein families modeled with hidden Markov models (HMMs), and Prosite motifs. A program scans the search results and makes preliminary name assignments. This is followed by expert annotator review of each gene to insure accuracy of identifications. Proteins whose specific functions can not be confidently assigned are designated "putative" or given a less specific family name. Proteins without significant matches to an HMM, motif, or other proteins are considered hypothetical. Running in the background is Small Genome Control, a program which automatically updates the database as changes are made to small genome data. TIGR continues to develop new tools to further increase the accuracy and efficiency of small genome annotation.

Annotation is the procedure by which raw DNA or protein sequence data is analyzed and the genes are identified by name and/or function. The wide variety of methods used to analyze sequence data has made annotation an increasingly precise process.

Gene Finding
The first major analysis step after a genome is sequenced is to identify the genes. The Glimmer system (Salzberg et al., 1998; Delcher et al., 1999) is used to find genes in bacterial, archaeal, or viral genomes. Glimmer relies on nothing other than the DNA sequence itself since it can be trained from raw sequence alone. In tests on numerous completely sequenced bacterial genomes, the system consistently finds over 99% of the genes in a fully automated fashion. The Glimmer system is freely available to nonprofit research institutions, and has been distributed to hundreds of sites worldwide (see http://www.tigr.org/softlab).

The algorithm at the core of Glimmer is an Interpolated Markov Model (IMM), which is a special type of Markov chain. A Markov chain captures statistical information about a sequence by computing the conditional probability P(x|S) that nucleotide x appears after sequence S. For example, a 5th-order Markov chain would compute the probabilities associated with {A,C,G,T} occurring after all 1024 possible 5-base sequences. By estimating these 4096 probabilities for both protein coding and noncoding DNA, an algorithm can later use them to score new sequences and decide if the new sequences represent genes. Because DNA encodes proteins using a 3-letter code, the most effective Markov chains are those that look at triplets or multiples of triplets; i.e., 2nd-order, 5th-order, and 8th-order Markov chains all work well because they are computing statistics based on codons, dicodons, and tricodons respectively. One limitation of Markov chain approaches is that higher-order models require exponentially more data for training, since they contain exponentially more probabilities that need to be estimated. From a

typical bacterial genome, one can usually extract training data sufficient to train a 5th-order model, but no larger. The advantage of IMMs is that they get around this limitation. Rather than estimating all probabilities associated with a fixed-length sequence (e.g., 5 bases), IMMs consider much longer sequences, but only use the statistics if enough data is available. Even when there is insufficient data to train an 8th-order model, there may be some short 8-base sequences (8mers) that occur frequently enough that a computational system could use them to estimate the probability of the next base. The IMM implemented in Glimmer works much like this: it calculates probabilities for all Markov chains from 0th-order through 8th-order, but only those for which sufficient data is available.

At first, training a bacterial gene finder on a newly sequenced genome might seem to present a chicken-and-egg problem. Fortunately, it turns out to be solved relatively easily through a simple automated procedure using code that comes with the system. Because bacterial genomes are very gene dense (about 90% coding, on average), a relatively trivial gene finder can find up to half the genes very quickly. The procedure works as follows: given a pre-selected length (for example, 600 base pairs), select all open reading frames (ORFs) of this length or longer. Then eliminate all ORFs from this set that overlap others in the set. The remaining "long" ORFs are likely to be genes, since they are too long to occur by chance and further they do not overlap other long ORFs. The best choice of length is a function of GC-content; it ranges from 500 base pairs to over 1000 base pairs for high-GC genomes. It should be noted that another training option is to run BLAST searches against a protein database, and to use all sequences with homology to another organism as training. This is quite effective, though very CPU-intensive. It is also possible to train on one DNA sequence and run the gene finder on another; this option has proven useful in several projects where only a small sample of DNA had been sequenced from a particular organism.

Once this training set is selected, the Glimmer system uses it to build an IMM. This IMM is then used by a separate program to scan the genome and predict all the genes, with criteria imposed for presence of an initiation codon and length of ORF. All the steps involved in running Glimmer, from training through gene finding, take less than 3 minutes on a desktop Pentium PC.

One of the limitations of the system is its inability in a few cases to resolve overlaps between genes; i.e., to decide what to do when two genes in different reading frames overlap. (There are six possible reading frames, three in each direction on the double-stranded DNA.) Because it is extremely rare for the locations of genes to overlap, the system tries not to permit this, and uses a fairly sophisticated set of rules to adjust the locations of the start codon to eliminate overlaps. Nonetheless, cases arise that are difficult to decide automatically, and in these situations the system outputs both genes along with a comment indicating that they need to be examined manually.

Functional assignment

Once the ORFs that are candidate genes have been chosen by Glimmer, several types of searches are perfumed on the set of hypothetical proteins they encode. Each protein is

searched against an internal non-redundant amino acid database (nraa) made up of all proteins available from GenBank (http://www.ncbi.nlm.nih.gov), PIR (http://pir.georgetown.edu), SWISS-PROT (http://www.expasy.ch/sprot) and TIGR's internal protein database, EGAD (http://www.tigr.org). The search algorithm employed for these searches is BLAST-Extend-Repraze (BER). This program first does a BLAST search (Altschul, et al., 1990) (http://blast.wustl.edu) of each protein against nraa and stores all significant matches in a mini-database. Then a modified Smith-Waterman alignment (Smith, 1981) is performed on the protein against the mini-database of BLAST hits. In order to identify potential frameshifts or point mutations in the sequence, the gene is extended 300 nucleotides upstream and downstream of the predicted coding region. If significant homology to a match protein exists and extends into a different frame from that predicted, or extends through a stop codon, the program will continue the alignment past the boundaries of the predicted coding region. The results can be viewed both as pairwise and as multiple alignments of the top scoring matches.

All of the proteins from the genome sequences are also searched against hidden Markov models (HMMs) using the program hmmpfam (Eddy, 1999) Two sets of HMMs are used: the Pfam HMMs (Bateman, et al., 2000), and TIGRFAMs (Haft, et al.). HMMs are built from highly curated multiple alignments of proteins thought to share the same function or to be members of the same family. They are useful for annotation since they are more sensitive and accurate than any pairwise alignment. HMM searches result in a score measuring the
probability that the query protein belongs to the group of proteins used to build the model. Each HMM has an associated cutoff score above which hits are known to be significant.

Several additional searches are performed on the proteins. Often there is duplication of genes within a genome resulting in families of proteins that are closely related to each other. These paralogous families are identified by searching the proteins from a genome against themselves using the program XDOM (Gouzy et al., 1997). Multiple alignments of these families are generated. In addition, searches for PROSITE motifs (Hofmann, et al., 1999) (http://www.expasy.ch/prosite), lipoproteins, signal peptides (Neilsen, et al., 1997), and membrane spanning regions (Claros, et al., 1994) are performed.

Non-coding genes and other features of the genome are also identified during the annotation process. The program tRNAscan (Lowe, et al., 1997)) is used to find tRNAs. Other software written at TIGR is used to detect structural RNAs, rho-independent terminators, and DNA repeats.

Once the searches are completed, we attempt to assign a putative function to each predicted coding region. TIGR has developed a two-stage annotation protocol whereby an initial automatic annotation is followed by manual curation of each gene assignment by TIGR annotators.

We have developed a computer program, AutoAnnotate, that analyzes the BER and HMM search results and assigns a common name, gene symbol, Enzyme Commission

(EC) number (http://www.expasy.ch/enzyme), and TIGR role category automatically. The program looks first at the HMM search results for a subject protein. If there is a hit to an equivalog-level HMM with a score above the trusted cutoff (an equivalog-level HMM describes orthologs that have conserved function as well as sequence), the identifying information attached to that HMM (common name, role category, gene symbol and EC number if applicable) is assigned to the predicted coding region. If no equivalog-level hit exists, the BER search results are evaluated. The program looks for a full-length match (at least 80% of the length of the subject) with a high percent identity (at least 35%). If more than one match is found, the program will attempt to choose a match with a name that
follows our naming conventions and assign a role. If the chosen BER match is a hypothetical protein from another species or if no pair-wise matches meet the match criteria, AutoAnnotate will go back to the HMM results and look for non-equivalog hits. If any hits exist, the protein will be assigned a family name based on the HMM name, for example, "transcriptional regulator, TetR family". Proteins with a pair-wise match to a hypothetical protein from another species, but no HMM hit, are named "conserved hypothetical protein". Proteins with no HMM or BER matches remain "hypothetical protein".

Following this initial "annotation", TIGR microbial annotators review each gene to insure that correct information has been assigned. We have developed an HTML-based graphical user interface that allows annotators to easily view all the data accumulated for each predicted coding region.

A main information page displays the identfying information (common name , gene symbol, EC number, TIGR role category) and summarizes results from the various search programs. HMM scores can be viewed and the user can follow links to internal and external pages that fully describe any particular model or to the multiple alignment of the predicted protein to the proteins that seeded the model. PROSITE motifs found in the predicted protein are listed along with the subsequence from the protein that matches the motif. One can also link to the PROSITE documentation at ExPASy. A table summarizing the BER search results links to a display of the pairwise alignments. Links within the alignment display take the user to the source database entry of the specific match. At these databases, annotators can view information about the match proteins such as active sites, membrane spanning regions, and DNA-binding sites. Annotators also check to see whether this information was experimentally determined or inferred from sequence similarity. Using this information, annotators can assess whether the protein from the predicted coding region has the same motifs, domains, and functionality as the match protein.

The information page also displays physical characteristics of the predicted coding region. The coordinates of the gene, the gene and protein lengths, molecular weight and pI are listed. The annotator can link to views of the DNA and protein sequences, membrane spanning regions, and a graphical view of the region surrounding the gene on the genome.

Distilling all of this information from a wide variety of sources to an accurate gene assignment is a complex task. We strive to annotate each gene with as much information as we can confidently impart, but are also wary of inferring too much from sequence similarity. This has led to a conservative approach to gene naming and a system of nomenclature in which the specificity of the name reflects our confidence in the assignment.

If there are multiple lines of evidence indicating that a protein has a specific function including HMM matches, multiple full-length pairwise matches with percent identity greater than 30%, and conserved PROSITE motifs (where applicable), then we use the fully descriptive name of the protein and assign a gene name. Some examples: a name reflecting high confidence is "ribose ABC transporter, permease protein (rbsC)". However, if the search indicates multiple sugars that could be the substrate for this transporter, a more general name would be given to the protein: "sugar ABC transporter, permease protein", without a gene name. Further, if the type of substrate that the transporter carried was not clear it would be named: "ABC transporter, permease protein". Finally, if even the type of transporter could not be confidently assessed, then it would be named "transporter, putative". In some cases, only membership in a defined protein family is confidently known, and in these cases we assign only the family name to the protein. In this way only as much information is included in the common name of the protein as can be confidently determined.

Each non-hypothetical gene is also assigned to a role category. The TIGR role category scheme was adapted from Riley (Riley, 1993). Categorizing the predicted coding regions from a genome by the roles they perform in the cell facilitates genome analysis. For example, at a glance one can tell which metabolites it is able to synthesize and which it is not, and whether or not the organism is photosynthetic or motile. Such categorization also facilitates comparison between organisms.

TIGR now also assigns Gene Ontology (GO) terms to all genomic proteins. (See GO tutorial for more info on GO.)

As the genome is annotated, information for specific predicted coding regions changes and the database is updated. If a 5' coordinate for a predicted coding region is altered, the DNA and protein sequence of the predicted coding region change, and thus the HMM and pair-wise searches will be impacted, as well as molecular weight, pI, etc. The Small Genome Control system tracks changes to the database and updates the searches and properties associated with the information that has changed. When data for a particular predicted coding region is changed, the name of the predicted coding region and the type of change made is stored in the change log table in the database. Each night, Small Genome Control looks at the change log table for each genome. If no changes have been made, nothing is done. If the genome sequence has been updated (perhaps because of frameshift repairs), a suite of programs updates such things as RNA searches, GC content data, etc.
If the coordinates for any predicted coding regions have changed, programs will be launched to update the DNA and protein sequences, molecular weight and pI, and

searches will be run to update signal sequence and transmembrane region information. The new protein sequence will also be searched against the internal database and the HMM library. Under this system, the data is kept current.

Coding Region Management

It is a generally accepted observation that genes do not overlap each other or RNAs in microbial genomes. After Glimmer analysis of the ORFs some overlaps remain and must be reviewed by an annotator. Sometimes it is possible to eliminate an overlapping region simply by correcting the initiation site of one or both of the predicted coding regions. Otherwise, one of the two must be eliminated. If one member of an overlapping pair is a hypothetical protein and the other has identifying information, the hypothetical protein is deleted. If both have identifying information, the evidence the annotation was based on is evaluated; usually one protein has very weak evidence and can be eliminated. If both are hypothetical proteins, the Glimmer scores determine which is deleted. In some cases, the overlap cannot be resolved, in which case both predicted coding regions are retained.

To ensure that Glimmer did not miss any ORFs, all intergenic regions of the genome sequence are searched against our internal non-redundant database using WU-BLASTX (Gish, et al., 1993). Regions that have database matches are examined to see if missed ORFs need to be inserted. Usually there is a small number of short ORFs that Glimmer did not identify. They are inserted into the database and annotated.

Potential frameshift mutations or start or stop codon point mutations identified during analysis of the pair-wise alignments must be resolved. These sequence discrepancies can arise not only from actual mutation of the DNA, but also from errors in sequencing or sequence editing. If an annotator identifies a potential mutation of this type, the laboratory is notified. The sequence coverage and quality is checked, and the sequence is either verified or a repair is suggested. Annotators will merge, split or extend the predicted coding regions based on the lab report. If a frameshift or point mutation is verified, "authentic frame shift" or "authentic point mutation" is appended to the common name of the gene.

Once annotation is complete, several automatic error checking programs are employed to identify annotation inconsistencies. The start and stop codons are checked to ensure that the coding region is correct. All genes that are not hypothetical or conserved hypothetical are checked to see that they have a role assignment. All genes are checked to ensure that no duplicate gene symbols are assigned. Coding regions that overlap RNAs are detected and
reported. In addition, a variety of internal database consistency checks are performed. When working with large amounts of information, it can be difficult to detect annotation errors. Common mistakes are easily caught by such automated procedures.

Data Availability

Once all of the annotation and coding region management steps have been completed for a genome and the work has been published, the genome is added to the Comprehensive Microbial Resource (CMR). The CMR is an interactive web-based tool, which allows

users access to information from every sequenced microbial genome at one site. There are tools available to investigate both data from an individual genome or data from many (or all) genomes. At this site, users can access the information used to annotate the genes, search for genes of interest, and compare their sequences to the genomic sequence. Also, users can download files containing the complete sequence of the genome, nucleotide sequences of each predicted coding region, protein sequences of each predicted coding region, and results of custom searches available on the site. TIGR also provides an email address for comments and questions.

Future Directions

We are continually trying to improve the quality of microbial annotation. A major effort is currently underway to expand our library of HMMs so that it will include every protein family. We are also establishing a flexible rule-based system that will assist in both annotation and consistency checks. Complex criteria can be combined in this system to improve accuracy. For instance, rules could be established so that a gene that hits the helicase HMM and has RuvB as its best BER match can only be named RuvB if there is also a gene identified as RuvA. Or, genes that are assigned the "photosynthesis" role category could be flagged for re-annotation in genomes from organisms that are non-photosynthetic. There is an obvious advantage in the ability to combine various types and sources of information when annotating, and the rules are easily updated as new information presents itself. Using these and other techniques, we hope to be able to annotate up to 75% of a microbial genome with minimal human intervention.

Acknowledgements

We wish to thank Jeremey Peterson, Erin Hickey, Robert DeBoy, Maria Ermolaeva, Granger Sutton, Tanja Dickinson, and the many members of the TIGR Bioinformatics staff, present and past, who worked on the tools described here.

References:

Altschul S., et al. Basic local alignment search tool. J. Mol. Biol., 215: 403-410 (1990)

Bateman A., et al. The Pfam protein families database. Nucleic Acids Res. 28(1): 263-266 (2000).

Claros M, et al. TopPred II: an improved software for membrane protein structure predictions. Comput.Appl.Biosci., 10(6): 685-686 (1994).

Delcher A.L., et al. Improved Microbial Gene Identification with Glimmer. Nucleic Acids Res., 27(23): 4636-4641 (1999).

Eddy S. Profile hidden Markov models. Bioinformatics, 14(9):755-763 (1998).

Gish W, et al. Identification of protein coding regions by database similarity search. Nat. Genet. 3(3): 266-272 (1993).

Gouzy J., et al. XDOM, a graphical tool to analyze domain arrangements in any set of protein sequences. Comput. Appl. Biosci. 13: 601-608 (1997).

Haft D., et al. TIGRFAMs: A protein family resource for the functional identification of proteins. Nucleic Acids Res. 29(1): 41-3 (2001).

Hofmann K., et al. The PROSITE database, its status in 1999. Nucleic Acids Res., 27: 215-219 (1999).

Lowe T., et al. tRNAscan-SE: a program for improved detection of transfer RNA genes in genomic sequence. Nucleic Acids Res. 25: 955-964 (1997).

Riley M. Functions of the gene products of Escherichia coli. Microbial Reviews, 57: 862-952 (1993).

Salzberg S., et al. Microbial Gene Identification using Interpolated Markov Models. Nucleic Acids Res., 26(2): 544-548 (1998).

Smith T.F., et al. Identification of common molecular subsequences. J. Mol. Biol. 147(1): 195-197 (1981).