INTERNSHIP REPORT

COMPUTATIONAL TOOLS FOR THE ANNOTATION OF NASONIA GENOME

An internship report presented in partial fulfillment of the requirement of the Professional Science Master's In Computational Biosciences

Prashanthi Selvanarayanan

Computational Biosciences Program

Arizona State University

Dr. Jürgen Gadau, Ph.D.
Dr. Stephen Pratt, Ph.D.
Dr. Zdzislaw Jackiewicz, Ph.D.

Internship Advisors

Arizona State University

NOT CONFIDENTIAL

Technical Report Number: 07-13

DATE: 11/30/07
ACKNOWLEDGEMENTS

I would like to express my gratitude for the support and valuable guidance of my internship advisors Dr. Jürgen Gadau and Dr. Stephen Pratt. I would also like to thank Dr. Rosemary Renaut for her valuable guidance throughout the program. I extend my sincere thanks to Dr. Oliver Niehuis and Josh Gibson for all their contributions.

I would also like to express my appreciation for the support and help of Dr. Ian Korf. I would also like to thank Dr. Zdzislaw Jackiewicz for kindly consenting to be on my committee and for his invaluable feedback. I am deeply appreciative of the support and guidance of Mrs. Renate Mittelmann. I am also extremely thankful for the support and love of my husband Vivek and my parents.
TABLE OF CONTENTS

TABLE OF FIGURES....................................................................................................- 4 -
ABSTRACT....................................................................................................................- 5 -
OBJECTIVES.................................................................................................................- 7 -
INTRODUCTION ..........................................................................................................- 8 -
IMPORTANCE OF THE NASONIA GENOME ANNOTATION .........................- 10 -
THE ANNOTATION SYSTEM...................................................................................- 12 -
ANAP - THE WORKFLOW SYSTEM .................................................................- 13 -
TOOLS USED IN ANAP .............................................................................................- 16 -
BLAST ......................................................................................................................- 16 -
GENE FINDING IN NOVEL GENOMES ..............................................................- 18 -
SNAP ....................................................................................................................- 21 -
GNOMON GENE PREDICTION ........................................................................- 25 -
FUNCTIONAL ANNOTATION..............................................................................- 26 -
GENBANK...........................................................................................................- 26 -
GBROWSE .............................................................................................................- 27 -
FUTURE DIRECTIONS ..............................................................................................- 30 -
BIBLIOGRAPHY.........................................................................................................- 32 -
APPENDIX A: PERL SCRIPTS USED.................................................................- 34 -
APPENDIX B: SHELL AND CGI SCRIPTS ..............................................................- 36 -
TABLE OF FIGURES

Fig. 1 Haploid Inheritance in *Nasonia* ................................................................. - 9 -

Fig. 2 (On the right) *N. vitripennis*; (On the left) Life Cycle of *Nasonia*............. - 10 -

Fig. 3 Design Graph of ANAP .................................................................................. - 12 -

Fig. 8 Implementation Diagram for ANAP ............................................................. - 13 -

Fig. 4 Protein Coding Gene ..................................................................................... - 18 -

Fig. 5 Gene Structure in Eukaryotes ..................................................................... - 19 -

Fig. 6 SNAP HMM State Diagram ........................................................................ - 22 -

Fig. 7 Multiple Software Modules at different layers in GBrowse......................... - 27 -
ABSTRACT

The Nasonia genus belongs to a diverse and important group of insects, the parasitic Hymenoptera. It is a genetic model for all parasitic wasps which are used in the biological control of insect pests.

The automated annotation tools presently available are not organism specific. Hence, applying an existing annotation pipeline to a new organism would be a complex process. Also the methods of gene prediction may not be precise. Hence developing a customized automated system would enable accurate gene prediction in the Nasonia genome.

This is achieved by developing some tools to provide accurate preliminary annotation for the desired sequences. The annotation system with a graphical interface integrates many commonly used bioinformatics tools. It also provides the user with a graphical visualization interface to aid in further manual annotation.

The automated program structure is similar to existing annotation pipelines with additional features and customized for Nasonia. An initial homology search using annotated genes from an already published genome against the genome of Nasonia is performed using BLAST. Gene Prediction is also performed using an ab initio gene finder which has been trained with the genome of closely related species. Bootstrapping methods are employed for parameter estimation to obtain species-specific parameters. Genome-Genome comparative alignment is performed using BLASTP and TBLASTN.
The predicted genes and those annotated at Genboree, maintained by Baylor College of Medicine, are currently displayed on a genome browser to enable subsequent manual annotation.
OBJECTIVES

The genome sequencing and annotation of *N. vitripennis* and new technological developments will allow the functional characterization of genes that can help to answer some fundamental questions in evolutionary genetics and applied entomology. The high throughput genome sequences combined with *in silico* functional annotation will help to identify phylogenetically conserved gene lineages that may provide baseline information for the improvement of parasitoids for biological control of arthropod vectors of human disease.

The educational objectives of the project are to increase my knowledge of comparative genomics, understanding of BLAST searches and limitations of alignment techniques. It also aims at improving my knowledge of gene finding techniques and principles involved in predicting them and to improve my programming skills in shell script and Perl.

The requirements of the project were

1) to determine steps that will be involved in the gene annotation system
2) to determine software and tools that will be required to implement the steps
3) to train the gene finding software in order to solve the purposes of the system
4) to develop scripts to implement the system
5) to identify the best visualization tool to display the annotated genes for purposes of comparative genomics
INTRODUCTION

The Parasitic Hymenoptera, also known as the parasitoid wasps are emerging as model organisms for complex genetic traits. They have diversified into a variety of life histories. They are crucial to agriculture and forestry as they attack some of the most important pests in the orders Diptera, Lepidoptera, Hemiptera, and Coleoptera at all life stages viz., egg, larval, nymph and pupal stages. Idiobionts and Koinobionts have been identified as the two main types of parasitoid life history. The idiobionts are ectoparasites while the koinobionts are endoparasitoids, i.e., the idiobionts kill or immobilize their host, ceasing its further development mostly at an immobile stage such as an egg or pupa while the koinobionts kill the host at a latter developmental stage, allowing its earlier development. The parasitoid wasps are also considered to be major regulators of arthropods and vectors of human disease like roaches, ticks and house flies.(Werren et al., 2004 )

Nasonia belong to the hymenopteran family Pteromalidae. They are small, 2-3 mm long wasps which parasitize the pupae of Calliphoridae and Sarcophagidae. For more than 50 years Nasonia, has been extensively studied in the field of genetics particularly genetics, evolutionary genetics, molecular evolution and comparative genomic research. It is the model for parasitoid genetics for the following reasons:

1) its large family size,
2) its ability to easily produce inbred lines
3) its ease of producing recombinant individuals that are genetically identical.
4) haplodiploid sex determination and parthenogenesis (see Fig. 1)
5) high recombination rate allowing positional cloning
6) short generation time
7) ease of handling

The above make *Nasonia* ideal for the study of complex genetic traits.

The genus contains three closely related species, *N. vitripennis*, *N. longicornis*, and *N. giraulti*. One difference between the species is in the time taken for development which is a little longer in *N. giraulti* and *N. longicornis*. Another contrast between *N. giraulti* and the other two species is that often females in *N. giraulti* mate within the host prior to emergence. The males in *N. vitripennis* have reduced wing size compared to the other two species.
Fig. 2 (On the right) N. vitripennis; (On the left) Life Cycle of *Nasonia* (WARD's Natural Science, )

**IMPORTANCE OF THE *NASONIA* GENOME ANNOTATION**

Parasitoids represent a source of biological control agents as they form important natural enemies of insect pests by keeping their populations in check. Today, pest management programs for agriculture and forest systems contain parasitoid species as an integrated component. Manipulating host preference, tolerance to surroundings and so on in parasitoids will help effective biological control of pest insects. Human health can benefit from the control of invertebrate vectors of disease, like the house fly. This can be achieved from the information obtained from the annotation of the *Nasonia* genome.(Werren et al., 2004)
Due to its phylogenetic position, the *Nasonia* genome may provide additional orthologs to human genes. Thus it can enable better understanding of the genetic architecture of complex traits and might ultimately lead to a better diagnosis and treatment of complex diseases. The annotation of the first genome sequence of a parasitoid species (*Nasonia vitripennis*) provides an opportunity to improve our understanding of important biological processes and increase benefits for human health. (WARD's Natural Science, )

The genome of *N. vitripennis* had been sequenced by the Human Genome Sequencing Center at the Baylor College of Medicine in Houston with funding from the NHGRI at NIH. It was performed as part of a project led by Jack Werren, University of Rochester, and the *Nasonia* Genome Sequencing Consortium. (*Nasonia genome project.*)
The primary reason behind the development of *Automated Nasonia Annotation Pipeline* (ANAP), a system for the annotation of the *Nasonia* Genome, was to deliver reliable and precise genome annotations using a wide range of evidence from different databases. This requires the integration of many bioinformatics tools available for automatic and efficient analysis and storage of genome data. The annotation system is designed such that genomic sequences pass through several successive levels with each layer of processing providing further refinement of annotation detail. ANAP involves three consecutive steps:

1. Raw sequence pre-processing by identification of homology matches using an alignment tool, BLAST
2. Gene predictions based on an ab initio gene finder, SNAP
3. Functional annotation based on comparative genomics
ANAP - THE WORKFLOW SYSTEM

ANAP is a web-based system designed to provide automated genome annotation followed by the visualization of annotated genes. It is based on a rule set of heuristics commonly employed in annotation projects. ANAP was thus created to facilitate the annotation of *Nasonia* genome sequences at Arizona State University. ANAP was
essentially was conceived in two parts: as an automatic annotation method and as a web-based genome display for *Nasonia* genome.

Automated genome annotation usually commences with running various stand-alone analyses. In ANAP, this initial stage of computation involves homology searches using BLAST. When a file is uploaded, the program creates a directory with the specified job name. Under this directory, the input file is placed under a subdirectory called Blast_input. Now, genes/proteins from other (related) species are then used to locate transcripts which have not been found previously. The results of the homology search are filtered using e-values. The system allows the user to input the e-value based on the user’s requirements. The results of the BLAST homology search are stored in a newly created subdirectory called Blast_output. Although, BLAST is a powerful tool for locating protein and cDNA sequences in the genome, it is not suitable for predicting gene structures. BLAST only detects homology and does not have a model for splice sites and exon boundaries. A sophisticated gene-finding strategy is required, where homology results can be extended and augmented so that an incomplete prediction can also yield an accurate gene structure. Information obtained from homologies is combined with information from other data sources to derive a full transcript structure. This adds a level of complexity to the automatic annotation process by meaningfully combining the data from independent analyses (Curwen et al., 2004).

The predicted genes (SNAP/Gnomon) are compared against the sequences resulting from homology matching. SNAP was trained for *Nasonia* before NCBI’s Gnomon predicted
genes for *Nasonia* were released. Hence the system gives the user the option of choosing the database of predicted genes. It is also incorporated in the system to enable cross-validation of gene-prediction.

This comparison yields two sets of outputs. The set of *Nasonia* sequences with homology matches which had significant sequence matches with the predicted genes is one of the outputs while the other is the set of sequences with no matches. These two outputs are stored under a new subdirectory called SNAP_output. The resulting files in SNAP_output undergo format conversion to make it suitable to be input to GenBank. The format converted files are stored under a subdirectory, Genbank_input. Putative gene function/functional annotation is based on BLASTX hits from the GenBank’s translated nr database. The results of the BLAST, constrained using e-vales input by the user are stored in a subdirectory called Genbank_output.

The web-based system allows the user to download the results from the server as well as receive it through email, which is specified by the user. If the user wants to view the output of the annotation in a genome browser, an option is provided for him to load it on the genome browser for comparative genomics purposes and further manual curation.
TOOLS USED IN ANAP

Different tools have been used together to provide better analysis of the data, which requires not only data transformation between tools, but also requires these tools to be seamlessly integrated with output of one tool to be input of another. Below is a detailed description of each tool that has been integrated into ANAP.

BLAST

Similarity search between sequences form an integral part of genome annotation because sequence similarity can reveal crucial information such as common ancestry and similar structure. Two common problems are over predicting genes and overlooking small exons. Over predicting genes means \[
\frac{\text{true positives}}{\text{true positives} + \text{false negatives}} = \text{sensitivity} = 100\%, \quad \frac{\text{true positives}}{\text{true positives} + \text{false positives}} = \text{specificity} \approx 0\%.
\]

In order to avoid this, BLAST, a powerful tool in homology search, has been used.

BLAST uses a heuristic algorithm that looks for local alignment as opposed to global alignments. Therefore, it is also able to detect relationships among sequences that share only isolated regions of similarity. The BLAST algorithm increases the speed of sequence alignment by first looking for \(k\)-tuples (common words) in the query sequence and each database sequence. The most significant words are sought and the significance is determined by evaluating these word matches using log odds scores in the BLOSUM62 amino acid substitution matrix, in proteins.
The word length, for the BLAST algorithm, is 3 for proteins and 11 for nucleic acids. The lengths ensure that a word score that is high enough to be significant but not long enough as to miss short but significant patterns. Low-complexity regions produce artificially high-scoring alignments which do not accurately convey sequence relationships. This is because they have many repeats of the same character or pattern and hence fewer sequence characters. Filtering these query sequences places emphasis on the more significant database hits. Low-complexity regions (marked with an X for protein sequences and N for nucleic acid sequences) are ignored by the BLAST program. The compositional complexity in a window of sequence of length $L$ is given by,

$$K = \frac{1}{L} \log_N \left( \frac{L!}{\prod n_i!} \right)$$

where, $N = 4$, for nucleic acid sequences and $N = 20$, for protein sequences

$K = 0$, for very low complexity

$K = 1$, for high complexity

$n_i$ = the numbers of each residue in the window. *(Using the basic local alignment search tool (BLAST) -- mount 2007 (14): Pdb.top17 -- cold spring harbor protocols.)*

The e-value for an alignment score (S) represents the number of hits with a score equal to or better than S that would be "expected" by chance when searching a database of a specific size. Aligned sequences with the expectation value above the expectation value cutoff are not considered while determining the conservation level for a position in the sequence. These sequences although listed on the BLAST Results tab do not affect the residues scoring. For gapped alignments also, the significance of a given alignment with
score S is represented by the e-value. This can be evaluated by looking at alignment scores generated using mock databases of random sequence of comparable length and composition. It decreases exponentially as the score (S) increases.

GENE FINDING IN NOVEL GENOMES

The process of identification of stretches of sequences which are essentially genes and are biologically functional refers to Gene finding. (See Fig for the structure of a protein-coding gene in eukaryotes.) For better understanding the genome of a species, accurate gene prediction is crucial.

One of the approaches to gene finding is based on the similarity of sequences/Homology-based. In this approach, a target sequence is searched to identify known genes that resemble the query sequence in a given a library of sequences of different organisms. The target sequence is also compared with expressed sequence tags of the same organism to identify regions corresponding to processed mRNA. The area of the target sequences that resembles the queries are putatively genes. Although, these approaches can find
biologically relevant genes, they could not identify genes that code for proteins not present in the library.

One of the most efficient approaches to finding genes in a genome is called the Ab initio approach. It is based on searching for certain signals of protein coding genes. However, *Ab Initio* gene finding is complex for Eukaryotes genes. The reason for this is 1) large intergenic regions separate the genes; 2) a gene is not contiguous, gene is divided into exons and introns by the splicing mechanisms. Defining open reading frames (ORFs) is difficult due to these splicing mechanisms. The signals are more difficult to identify since these signals are more complex and unspecified. Examples of such signals are start and stop codons, intron/exon boundary, binding sites for a Poly-A tail and CpG islands.

According to the first-order Markov chain, it is a linear series of states, in which the current state depends only on the previous state in the chain. Each state defines one or
more emissions with some probability associated with each occurrence. According to the Hidden Markov Model (HMM), if there are only the emissions from a Markov chain but not the underlying states, the states are then called hidden. Eg., Markov chain describing stretches of GC-rich and AT-rich DNA would have two states:1) G or C emission with high probability; 2) A or T emission with high probability). It is then assumed that the probability of staying in the same state is high at each step along the chain but switching occurs occasionally.

HMMs are used for finding Eukaryotic genes. They output the most probable hidden state path $S$ that generates the observed genome $G$ using Viterbi algorithm for a given genome $G$ of length $L$. The probability of the hidden state sequence $S$ given $G$ is computed using Bayes’ rule(Durbin, Eddy, Krogh, & Mitchison, 1999):

$$P(S|G) = \frac{P(S,G)}{\sum S' P(S',G)},$$

$S'$ is in the set of all the possible hidden state path of length $L$.

A simple application of HMMs to de novo gene prediction assumes that the observed states are nucleotides of the target sequence and the functions they serve in RNA processing and translations are the hidden states. Eg., a base in the middle region of an intron, the first and second base of an intron or the first, second and third base of a codon. The observation corresponding to each state of a HMM is always a single nucleotide.

One of the primary and key tasks in a new genome is to determine the structure of its protein-coding genes. Accuracy of a gene finder relies on proper training but training a gene finder is a difficult task. At present, complete genome sequencing assembly is so
rapid that genomes appear much before proper training material appears. As a result of this, some genome annotation projects do not have an appropriate gene finder. New genomic sequences are annotated with a gene finder for a completely different genome. Little training material and experimental data may be obtained for a newly sequenced genome to anchor gene predictions. Finding genes in an uncharted territory such as this can be laborious. Hence, a gene finder from the most phylogenetically similar genome is commonly used. But, annotations using a foreign gene finder could be highly inaccurate.

However, picking the best foreign gene finder is not just a matter of using parameters from the closest relative. In order to train a gene finder for a novel genome in the absence of any data, the predicted genes should have compositional properties similar to the actual/real genes. Instead of using foreign gene finders to identify genes, using them to bootstrap parameter estimation would yield more accurate annotations.

**SNAP**

Semi Hidden-markov-model Nucleic Acid Parser (SNAP) is a high-performance ab initio gene finder. It models protein coding sequences in genomic DNA using a specialized hidden Markov model. The model of semi-Markov type, is formulated as an explicit state duration Hidden Markov Model (HMM). The model is thought of as generating a ```parse``` 

\( \varphi \), consisting of an ordered set of states, \( \tilde{q} = \{q_1; q_2 \ldots q_n\} \), with an associated set of lengths (durations), \( \tilde{d} = \{d_1; d_2; \ldots ; d_n\} \) which, using probabilistic models of each of
the state types, generates a DNA sequence $S$ of length $L = \sum_{i=1}^{n} d_i$. The generation of a parse corresponding to a sequence length $L$ is as follows:

1. An initial state $q_1$ is chosen according to an initial distribution on the states, $\pi$ i.e., $\pi_i = P\{q_1 = Q^{(i)}\}$, where $Q^{(j)} (j = 1, \ldots, 27)$ is an indexing of the state types.

2. A state duration, $d_1$, corresponding to the state $q_1$ is generated conditional on the value of $q_1 = Q^{(i)}$ from the length distribution $f_{Q(i)}$.

(3) A sequence segment $s_1$ of length $d_1$ is generated, conditional on $d_1$ and $q_1$, according to an appropriate sequence generating model for state type $q_1$.

3. The subsequent state $q_2$ is generated, conditional on the value of $q_1$, from the state transition matrix $T$, i.e., $T_{i,j} = P\{q_{k+1} = Q^{(j)}| q_k = Q^{(i)}\}$

A variant of the above semi-Markov type algorithm has been used in SNAP. The significant differences are that SNAP uses six intron states to prevent stop codons at splice junctions. SNAP models each strand independently to enable genes on opposite
strands to overlap. This allows genes within introns of other genes, which is the case in some genomes it also allows overlapping exons, which is not common and is sometimes disadvantageous. A parameter file specifies the state diagram in SNAP. Changing the HMM to describe a variety of genomic features is thus allowed (See Fig. 6 for the simplest state diagram for predicting single- and multi-exon genes). Any length weight matrix and any order Markov model can be employed and to embed the sequence feature models within a coding sequence framework or an array or even decision tree.

**PARAMETER ESTIMATION:**

Parameter estimation from gene predictions would be an easy and convenient way to train a gene finder for a novel genome. The bootstrapped parameters in genomes can rival actual data, and even the worst bootstrapped parameters have been found to be reasonably accurate.

Training sequences are the sequences that the model should fit well with. Let these be \( x^1 \ldots x^n \). They are assumed to be independent and thus the joint probability of all sequences given a particular set of parameters is the product of the probabilities of the individual sequences. While working in log space, the log probability of the sequences,

\[
\log P(x^1 \ldots x^n | \theta) = \sum_{j=1}^{n} \log P(x^j | \theta)
\]

where \( \theta \) represents the entire current set of values of the parameters in the model(Korf, 2004).
For HMM gene prediction from genomic sequences, estimation of the probability parameters from known state sequence is required. When all the paths are known, the number of times each particular transition or emission is used by the set of training sequences can be counted. The maximum likelihood estimated for $a_{kl}$ and $e_k(b)$ are given by

$$a_{kl} = \frac{A_{kl}}{\sum_{l'} A_{kl'}}$$

and

$$e_k(b) = \frac{E_k(b)}{\sum_{b'} E_k(b')}$$

To avoid problems of overfitting, in case of insufficient data, it is preferable to add predetermined pseudocounts to $A_{kl}$ and $E_k(b)$. The pseudocounts should have a natural probabilistic interpretation on the probabilities for each state and should be positive.

**TRAINING SNAP FOR ANAP**

Initially SNAP, the ab initio gene finder, had to be trained for homology gene matches using genome sequences from *Apis mellifera* (Honeybee). An important decision in the annotation process is the choice of data sources to bootstrap parameter estimation. *A. mellifera* was chosen in this case because *Nasonia* is closely related to *A. mellifera* i.e., it is 120 million years ago diverged from *A. mellifera*, in a second major branch of the Hymenoptera.
After training SNAP, the predicted set of genes is used to retrain (bootstrapped) SNAP for parameter estimation. The resulting genes predicted by SNAP are stored in a file which is used in ANAP.

**GNOMON GENE PREDICTION**

Gnomon is the technique used at NCBI for gene prediction. It finds the maximal self-consistent set of corresponding transcript and protein alignment data using a set of heuristics in order to set the constraints for an HMM-based gene prediction. (*Gnomon description.*)

The coding propensity of the available transcript alignments is evaluated to determine the most probable coding regions. After choosing single set of non-overlapping transcript alignments with better coding propensity, the best matching proteins for these transcript alignments are aligned back on the genomic DNA sequence.

In the case of transcript alignments, the program ensures that the chosen coding region is part of a putative mRNA. For the protein alignments, the predicted gene has to have every exon in the right frame as suggested by the protein alignment. However, the program is free to introduce other exons between parts of the protein alignment and choose the splice sites.
FUNCTIONAL ANNOTATION

The key to understanding biological systems of an organism is the functional annotation of its genes. An annotation can be defined as a gene product which:

1) has a specific molecular function
2) is involved in a particular biological process
3) is located within a particular cellular component

An approach used extensively in protein function annotation is the comparative genomics approach. In this approach, annotations are based on the concept of conservation of genes across genomes. The observation that leads to this concept is the genes and functional elements mutate at a slower rate than the other parts of a genome due to the force of natural selection. Gene prediction can be achieved by comparing the novel genomes against genomes sequenced in related species to detect this conservation.

GENBANK

It is an open access sequence database produced at NCBI as part of the INSDC. It contains annotated collections of all nucleotide sequences and the translated protein that are publicly available. Its growth is exponential as it doubles every 10 months. It contained more than 67 million sequences, according to a release 158, produced in February 2007(GenBank overview.). GenBank entries comprise submissions from individual laboratories, and also bulk submissions from large-scale sequencing centers. These submissions can be done using the web-based BankIt or Sequin, a stand-alone submission tool. There is exchange of data with both the EMBL Data Library and the
DNA Data Bank of Japan everyday to ensure not just up to date data but also global coverage.

GBROWSE

GBrowse is a web-based display that can display an arbitrary set of features on a nucleotide or protein sequence, along with dozens of aligned annotation tracks.

GBrowse consists of a CGI (Common Gateway Interface) script for managing the user interface, at the top level. This script provides the end users with forms to interact with, accept and process requests. It also preserves users’ preferences for sessions by managing the cookies and displays the rendered images of annotated regions.

Two software libraries support GBrowse – 1) Bioperl library; 2) Bio::Graphics software library. The bioperl library provides interface between the CGI script and the underlying database while the Bio::Graphics module renders the genome images. Beneath the libraries is a relational database which stores and retrieves features. (Stein et al., 2002)
CONCLUSIONS

The goal of this project was to create a genomic pipeline for the annotation of *Nasonia* genome. The annotation system has been implemented using Blast, SNAP and gbrowse. The system has an easy to use interface that allows users to interact with the system to enable the system to be robust and perform efficiently.

The annotation of the *Nasonia* genome will open the door to better understanding of important biological processes. It will also lead to methods for further enhancing their utility for control of agricultural pests and disease vectors and thus benefiting human health.

The project also helped me to further my understanding of Biological sciences while improving my existing computational skills. With respect to biological sciences, I learnt how genes are predicted and annotated, importance of gene annotation, the heuristics behind it and the complexities of a genome.

With respect to computation sciences, my programming skills have improved in Perl, CGI and shell scripting. It gave me an opportunity to install, manage and maintain apache web-server and MySQL database server. This project also allowed me to put my existing UNIX to work, while improving my knowledge of Mac OS.
With respect to bioinformatics tools, the project has given me an opportunity to work with BLAST, GBrowse and usage and customization of ab initio gene predicting software called SNAP, FgeneSH and GeneZilla.
FUTURE DIRECTIONS

The possible directions the project can take in the future are:

1. Expand the capability of the tool to other species.

   The tool currently is customized only for *Nasonia* Genome. The tool can be made
generic to support other genomes. This can be achieved by making the tool
flexible to accept user customization for:

   a. BLAST database
   b. Gene prediction

   This would be a very good improvement to the project as ANAP has proved to be
useful tool in the annotation of *Nasonia*. Hence it would be useful in cases where
a novel genome is being annotated.

2. Improve the gene predicting capability of the tool.

   The crucial task in the tool is gene prediction. Improving the efficacy of the tool
in this area would be very critical to the objective of the project. The gene
predicting capability of the tool depends on the gene comparison program and the
predicted genes derived from SNAP.
The improvement can be achieved by:

a. Updating the predicted gene output from SNAP regularly. The SNAP predicted genes can be updated as and when genes in *Nasonia* are annotated. Using these genes to bootstrap ANAP will allow better prediction resulting in more accurate annotations.

b. Using a more customized gene predicting software.

Currently, there is no customized gene predicting software available for *Nasonia*. This was the fundamental reason for choosing SNAP. As *Nasonia* genome annotation progresses, more customized software could be made available. The customized tool can then be used to predict genes and the output can be used instead of SNAP/Gnomon.
BIBLIOGRAPHY


generic genome browser: A building block for a model organism system database.

*Genome research, 12*(10), 1599-1610.

*Using the basic local alignment search tool (BLAST) -- mount 2007 (14): Pdb.top17 --

http://www.cshprotocols.org/cgi/content/full/2007/14/pdb.top17


http://resources.wardsci.com/tag/nasonia/


*Proposal to sequence the nasonia genome.* Retrieved 10/15/2006 from

http://www.genome.gov/Pages/Research/Sequencing/SeqProposals/NasoniaSeq.pdf
APPENDIX A: PERL SCRIPTS USED

multiple2single.pl
This script separates a set of FASTA-format sequences in the input file into individual files. The name of each new file is the name of the contig.

get_gene_hits.pl
This script compares the blast output with the genes predicted by SNAP/Gnomon genes and creates two files as the output. One file contains the set of sequence corresponding to the blast output which has no hits in the predicted genes and the other contains the sequences corresponding to the blast output which has hits in the predicted genes.

Convert4gb.pl
This script converts the output of the comparison step to a suitable input for the GenBank BLAST step of the system.

Genbank_submit.pl
This perl script takes Fasta sequences as input and submits them to GenBank for performing BLAST search against nr database. On submission to GenBank, the program receives an request ID and a waiting time. The program waits for the specified waiting period and queries Genbank with the Request ID. GenBank either sends the response or requests the program to wait longer. This continues till the program receives the BLAST.
search output from GenBank. The program then parses the received response and writes the output to a BLAST output file.

Convertzff2gff.pl

This perl script converts GFF format files to ZFF format (Format used by SNAP) files for purposes of training and bootstrapping SNAP.
APPENDIX B: SHELL AND CGI SCRIPTS

ANAP
This shell script implements the workflow system. It calls the appropriate tasks used in the workflow sequence. The sequence of programs being called by ANAP is multiple2single.pl

SNAP_train
This shell script calls the perl scripts to convert the GFF format to ZFF. The script then trains SNAP using the ZFF file generated. SNAP is then bootstrap parameter estimated to predict accurately.

anap.cgi
This is a CGI script that takes the user input through the web and executes the ANAP workflow system. This script also sends the user email, once the workflow is completed.