CSE182 Class project: An EST database of *H. medicinalis*

October 15, 2006

1 Introduction to Hirudo

Hirudo medicinalis (*medicinal leech* is organism with historical medical as well contemporary relvance as a model organism for medicine. Our project will be on extracting biological information from a database of *transcribed* genomic (ESTs) sequences, and their applicability to Professor Eduardo Macagno's neuroscience lab at UCSD.

See *http://animaldiversity.ummz.umich.edu/site/accounts/information/Hirudo_medicinalis.html* for basic physiological information. It is a segmented annelid, with 21 body segments (basic plan is a theme with variations), 32 metameres (& neuromeres), seg ganglia with $\tilde{4}00$ neurons, neuronal function, morphology and connectivity characterized, good regeneration, innate immune response, medically important system, etc.

For neurobiology, H.m. is the only organism that is sufficiently well characterized as a whole. Most of its central neurons are identified and their functions/synaptic connections known (or can be easily studied), making it possible to relate detailed gene expression profiles to physiological programs for each neuron. For medicine, many important molecules/factors remain to be discovered. For evolution, it represents a clade (lophotrochzoa) and phylum (annelida) that is not well described. It is also the source of a number of bio-products, including anticoagulants, possible treatment for arthritic joints, useful as an adjunct to certain surgeries.

The basic questions facing the *Hirudo* community are:

1. What patterns of gene expression underlie neuronal function and synaptic circuitry?
2. What are the relations among gene expression programs related to developmental, regeneration, immune and other stress responses?
3. What leech specific factors (anticoagulants, proteases, antibiotic peptides, etc.) are of medical interest? How do synaptic circuits generate specific behavioral responses?

How does an EST resource help? Key to assaying expression profiles to correlate with function for individual identified neurons (either via in situ hybridization or gene microarrays), entry point for discovering genetic circuits, source of data for analysis of mass spec data and proteomics, source of partial transcripts that can be used to obtain full transcripts and protein sequences for siRNA, ectopic expression, biochemical pathways, etc. What questions do we hope to answer with an EST database? Functional questions about the origins of cell properties, evolutionary issues

2 Introduction to EST sequencing

We will discuss this, and related topics in class lectures. See a basic introduction in *http://www.ncbi.nlm.nih.gov/About/primer/est.html.*

3 Projects

The class projects are designed to produce data that is useful for the neuroscience lab. Therefore, your *customer* is Professor Eduardo Macagno, and members of his lab, who will answer any questions about the relevance of the project and their needs from the database. Your *consultants* on these projects are the instructor, the TA (Julio Ng), and Jeff Wang.

Please note that not everything described below might make sense to all of you, but the classroom lectures prior to the first mid-term should help with that. You must choose one of the following projects, working in teams of size 3 (except for project 4, where larger teams are allowed).

The project has 3 checkpoints, with dates on the course home page.

1. Checkpoint 1: Form teams, and decide on a specific project. download data.
2. Checkpoint 2: See individual projects for C2 requirements.
3. Checkpoint 3:

3.1 Functional analysis (annotation) team

Build a pipeline that does the following:

Input: A multi-fasta file of DNA sequences. The sequences can be individual ESTs or assembled transcripts. The header line of each entry is free-format but contains a unique ID.

Output: Search each sequence against each of the annotation (functional domain) databases mentioned below. For each type of 'annotation-source', produce a tabular (tab-delimited) output, with the following columns in each row:

1. Sequence Id
2. Annotation Id
3. Seq begin (The sequence coordinate where the domain match begins)
4. Seq end
5. Domain begin (relevant if the sequence matches only a portion of the domain).
6. Annotation end
7. P-value, Score
8. Strand (+/-), Frame (1,2,3) if relevant.
9. Domain specific fields.

Various annotations must be produced, including, but not limited to the following annotation-sources:

1. BLAST: Download the NCBI toolkit (ftp://ftp.ncbi.nih.gov/blast/)to run Blast locally, and download the NCBI nr database. Query each sequence as against the NCBI nr database. The annotation here is a simply a protein sequence. Use a conservative p-value cut-off after some experimentation.
2. psi-Blast. Exactly as for Blast.
3. PFAM: Download the Pfam database, and HMMer software toolkit from (http://pfam.wustl.edu/). Search each sequence for a Pfam domain using HMMer.
4. PDB: Download the PDB sequences (but not structures) from (ftp://ftp.rcsb.org/pub/pdb/derived_data/pdb_seqres.txt). Use Blast to search the sequences, and report possible structures if any.
5. UTR: Run the UTR prediction tool (XXX) to predict the coding region, frame and untranslated region of the EST. As there is no annotation Id, simply use "UTR" as the annotation ID.
6. PROSITE: download the prosite regular expressions from ftp://ca.expasy.org/databases/prosite. Search each sequence against the prosite database using either your own tool, or a tool like ScanMotif.

Checkpoint 2 deliverable

Produce annotation table for BLAST using all available EST sequences and assemblies. Support the Bioiformatics Algorithm Team I by providing them with the annotation table for *queryIndex*.

3.2 The web team

The goal of this project is simple: Create 3 views (dynamic HTML, or appropriate technology), and link them to the appropriate CGI scripts etc.

1. The Home page.
2. An EST-Collection Page
3. EST Sequence page.
4. EST Assembly page.

The Home Page

Should have a logo for the database (Choose an appropriate name (EX: Leechee (if you like the fruit), and design a graphical logo)), and a single search box. The search box is free-format, and it can accept EST-IDs, keywords (EX: annexins), or a sequence.

The output of the search is always be a collection of EST-assembly sequences, a collection of EST-sequences or both. If the output is a collection, display the EST-collection page. If the output is a single assembly, or a single EST, display the appropriate EST-assembly, or EST-sequence page.

To execute the query, you will need to do one of the two things:

1. Keyword/Id query. Use *indexQuery* to retrieve the sequence IDs.
2. Sequence query. Use BLAST to retrieve the sequence IDs.

EST-collection Page

There are two sections: Assembly, and EST. Under each section display the appropriate hyper-linked collection of EST-IDs, or Assembly-IDs.

EST-sequence page

Given an EST-sequence ID, this page is displayed. You will write a program that dynamically generates the following HTML page after parsing through the functional annotation files. As a generic format, please see GeneLynx, record#845 (http://human.genelynx.org/cgi-bin/record?glid=845&submit=Go). The page must have the following information:

1. The sequence.
2. 5' or 3' EST.
3. Amino-acid translation in the appropriate frame, the frame, and the strand.
4. Assembly ID. The ID of the assembly that contains this EST, hyperlinked to the EST-assembly page.
5. Domain Analysis: Use the tables generated by the annotation team to link this sequence to entries in PDB, Pfam, Prosite, and PDB. For each entry, display also the score/confidence of the annotation (obtained from the annotation team tables).

EST-assembly page

Given an EST-assembly-ID, this page is displayed. It is identical to the EST-sequence page, but will have additional sections. For each assembled sequence, display EST-sequence IDs hyper-linked to their own pages. Also, display a graphical view of how the ESTs are laid out to get the consensus assembly.

Checkpoint 2 deliverable

You should have the home page up and running and using the *queryIndex* supplied by the Algorithms I team. For the EST-sequence, and the EST-assembly pages, a Domain analysis of at least the BLAST results should be supported.

3.3 Bioinformatics algorithm team I: Index Team

The goal of this project is to create an index for supporting keyword and other queries. Create the following programs:

1. *keywordTrawler:* Download text dump of all the annotation teams databases (NCBI nr protein, Pfam, PDB, PROSITE). Coordinate with the annotation team to ensure that the same version is used.

   (a) For each annotation-source, build a table of keywords. Each row of the table contains a keyword and the unique ID of the entry from which the keyword was selected.

   (b) Discard 'common' words (words that occur with a very high frequency. For extra credit, also create 'phrases' of words that appear together, and index them as well.

2. *queryIndex:* This program takes as input a keyword, or a lst of keywords and returns EST-sequence IDs, or EST-assembly IDs associated with the keyword. For extra credit, also implement an OR option for multiple keywords.

   Note that *keywordTrawler* creates an index of keywords with an annotation ID (i.e., given a keyword, you can identify all the specific annotations that contain that keyword). The annotation team has tables that link the annotation ID with EST-sequence ID, and EST-assembly ID. Create a *join* of these tables to build queryIndex.

As the indexing programs use database concepts, one simple option is to use MySQL or similar relational database program to populate these tables, create indexes, and join. However, credit here will be given for writing your own code to build fast index queries. While this project is somewhat open ended, there are many tricks you can use to improve the results. These will give you extra-credit for the class to help boost your grade. Among the desirable characteristics are:

1. Use of an appropriate data-structure to query the index. Think how fast Google can search with its Google desktop index. What is the memory footprint of your index? Can a standard computer perform the search?

2. Can you correct simple typographical errors? At the very least, the search should be case-insensitive and strip out all punctuations. If possible, you should search for multiple forms of the same word.

3. If the user inputs multiple keywords, the default is to return entries that contain all of the keywords. However, if no such entries exist, the query should return entries that match at least some of the keywords, and use a scoring criterion for the hits.

4. Can you search for phrases, instead of words, like in Google.

Checkpoint 2 deliverable

For the C2 phase, implement and run *keywordTrawler* to search a protein sequence database (NCBI nr) and link it to accession number, or the gi number (coordinate with the Annotation Team). Write a simple version of *queryIndex* which allows single keyword queries. Support the web-team by designing a simple interface that allows them to use *queryIndex*. Show basic timing results. The requirement for the final project will be handed out after Midterm I.

3.4 Bionformatics algorithm team II: EST clustering and assembly

Note: This is a challenging programming project. You can work in teams of 5, and are allowed to to not do one of the assignments.

The input is a collection of ESTs. The goal is first to cluster ESTs so that each gene is represented by a single cluster, and second, to assemble the cluster sequences into a single consensus sequence. The output of a project will be a collection of Assemblies, represented in two files. The first is a file of sequences (*assembly.fa* in multi-fasta format, with a unique Assembly ID for each sequence. The second is a data file *assembly.dat* with the following information: For each assembly id, the file has a number of lines, each containing one of the component ESTs, with the following information:

1. EST sequence ID.
2. 5'/3'
3. begin-coordinate of the mapping of the EST to the assembly.
4. end-coordinate.

The project has many steps, and it is strongly recommended that students speak with the instructor during the project to esure that they stay on track.

1. Fast-all-against all comparisons. The first step in clustering is to compare each pair of ESTs to determine if they should be part of the same cluster. However, this is a significant computation, and you should use a hashing technique to speed it up.
2. 3' EST clustering. Details will be forthcoming.
3. 5' extension and layout. Details will be forthcoming.
4. 3'/5' EST assembly. Details will be forthcoming.
5. consensus sequence. Once you have the 3' and 5' clusters, you can use Phrap to get the data.
6. interesting isoforms

For Checkpoint 2, you must show a fast all-against all comparison. Your output should be a collection of ESTs, with pairwise BLAST scores, and statistics on the time and memory requirements of the clustering.

3.5 Mining the EST database for specific families

This project does not require as much programming, but you are expected to learn the relevant biology to make progress. You are encouraged to contact lab members from Macagno lab: Dr. Orit Shefi (oshefi@ucsd.edu) (Netrins/Receptors), Dr. Alejandro Sanchez (Innexins) (a2sanchezgonzalez@ucsd.edu), and Dr. Michael Baker (mwbaker@ucsd.edu) (HmLAR2). Note that there is considerable overlap with the functional annotation project. However, the rationale is that this project is about careful analysis of a few select families with experts in the Macagno lab, and any discoveries you make will be followed up. The functional analysis project is more a high-throughput effort, and will be rated for robustness, efficiency etc. I expect that 1-2 people can do a family, so you can work in larger team that shares code etc. but indviduals should be responsible for each family. Most examples given here are w.r.t Innexins, but please contact individual lab members to get advice on the family you have picked.

The goal of this project is to mine the EST database for all members of the following protein families which are studied by the Macagno Lab.: Innexins (The Gap junction proteins), Netrins (small, secreted proteins that help direct axon migration), and phosphatases (receptors and cytoplasmic, including HmLAR2, a tyrosine phosphatase receptor). The output of this project is:

1. A list of all EST, and Assembly IDS of sequences that belong to either of these families.
2. For each family, a phylogenetic tree of all known full-length members of that family from model species (C. elegans, Dros. and others).
3. Identify the orthologs of each Hirudo sequence in the other species, and write a review on the possible function of the orthologs.

There are a number of ways to identify (remote) homologs, so the following is simply to get you started.

1. Create a database of all known innexins, and use Blast to search the Hirudo database for instances.
2. Use known representations of the family (BLOCKS, Pfam, Prosite) to search the Hirudo database. For example, the Pfam entry PF00876 is an HMM representation for innexins. Points will be given for every novel sequence that Blast could not identify.
3. Use Psi-Blast.

To construct phylogenetic trees, you can use available programs from the PHYLIP package. Compare your results to the tree of 123 sequences in Pfam. For this project, it would be sufficient to build your tree using C. elegans, and Drosophila Innexin sequences. Next, introduce the Hirudo sequences onto this tree. Is any clade of the tree over-represented?

The following references might be useful as characterizations of Innexins in Drosophila:

Intercellular Communication: the Innexin Multiprotein Family of Gap Junction Proteins.
Chemistry & Biology, Volume 12, Issue 5, Pages 515-526
R. Bauer, B. Loer, K. Ostrowski, J. Martini, A. Weimbs, H. Lechner, M. Hoch

Starich T, Sheehan M, Jadrich J, Shaw J.
Innexins in C. elegans.
Cell Commun Adhes. 2001;8(4-6):311-4.

For checkpoint 2, you should show your preliminary list of ESTs/Assemblies for each family, and the 'evidence' for including them.

4 Possible Paper Outline

1. Introduction to *Hirudo*.
   (a) What are the important questions facing the Hirudo community?
   (b) Why is *H. medicanils* an important model organism for Hirudo development? Historically, what are the important discoveries that have been made on this organism.
   (c) How does an EST resource help?
   (d) What questions do we hope to answer with an EST database?
2. The Hirudo EST resource
   (a) Statistics: number of ESTs? Number of 3', 5' ESTs?
   (b) Number of clusters? Assembled sequence? Sequence coverage of known genes.
   (c) Interesting Repeat families? Alternative splicing?
   (d) Abundance curve of clusters.
   (e) Comparison to the Hirudo genomic sequence.
3. The Hirudo EST Proteome
   (a) What is the number of genes in Hirudo
   (b) Describe the Hirudo proteome, with abundance of different families. Compare the proteome against the proteomes of related species. Do any gene families expand or contract? is this significant?
   (c) Phylogenetic analysis of larger families. Is that possible with ESTs?
   (d) Differential expression between early development and adult.
4. ncRNA
   (a) What kind of ncRNA motifs are to be found in UTR regions?
   (b) Are some of the mRNA transported using 'zip-codes'?
   (c) Other ncRNA, miRNA.
5. Case Studies: examples of ESTs that were used to derive a full length clone on which interesting Biology was performed.