Analysis of pre-genome assembly of *Lymnaea Stagnalis*

Jonathan Neuteboom

August 8, 2011

Contents

1 Introduction

In 1977 the first DNA-based genome was sequenced [1]. This was the beginning of the sequencing of thousands of organisms. All this sequence information was used to analyze genes and decode proteins or other regulatory chemical compounds that are encoded by the genes, the genetic instructions vital for the development and functioning of every living cell. With, the growing amount of data, it became impractical to analyze it by hand, so more and more computers became an important part of the analysis. Today, computer programs, like *BLAST*, are used daily to search from over $260,000$ organisms containing 180 billion base pairs. This application of computer science and information theory to the field of biology is called Bioinformatics. The aim of Bioinformatics is to better understand the biological processes that are active in the cells of every organisms, using computational power. There are many research directions in Bioinformatics, including sequence alignment, gene finding, genome assembly, drug design, drug discovery, protein structure alignment, protein structure prediction, prediction of gene expression and protein–protein interactions and the modeling of evolution.

In this paper, the analysis of the DNA of the pool snail, *Lymnaea Stagnalis*, will be handled. Both quality and the information that can be retrieved from the data set, will be dealt with. Because we don't expect the reader to know much about the knowledge of DNA and its aspects, this will be explained in the following chapters.

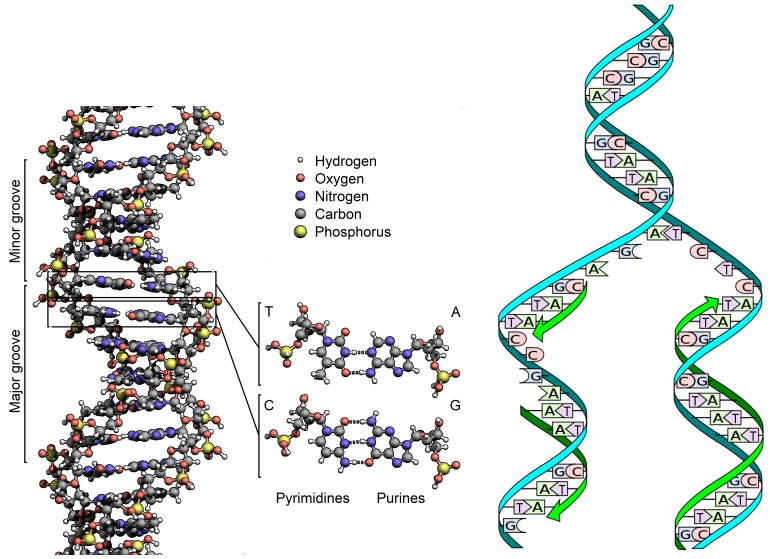
To analyze DNA, much background information is needed. To provide this information, the following topics will be covered:

- Genomes & DNA
- Sequencing
- Genes
- Blast
- Blast2GO

Subsequently, the definitions of the problems will be stated. The setup of the project will be discussed in Section 2, Section 3 will report the results and Section 4 will try to answer the problems and discuss the final results.

1.1 Genomes & DNA

The genome is all the hereditary information of an organism [2]. A genome consists of one or multiple chromosomes, which consist of DNA. DNA, short for Deoxyribonucleic Acid, is the building stone of all living organisms (except for RNA-viruses)[3]. It consists of two helical chains, coiled around the same axis, as shown in Fig. 1(a). Each chain consist of a sequence of nucleic acids. There are four different kinds of these building blocks: **A**denosine, **T**hymine, **G**uanine and **C**ytosine. Because they appear frequently in a strand (a chain of DNA can contain several millions of nucleic acids), they are abbreviated by **A**,



(a) The structure of DNA showing with detail the structure of the four bases, adenine, cytosine, guanine and thymine.

(b) DNA copying. The double helix is unwound and each strand acts as a template for the next strand. Bases are matched to synthesize the new partner strands.

Figure 1: Structure representation of DNA.

T, **G** and **C**, respectively. A second property of DNA is that each nucleic acid has a complement version (denoted C):

$$C(\mathbf{A}) = \mathbf{T}, \quad C(\mathbf{T}) = \mathbf{A}, \quad C(\mathbf{C}) = \mathbf{G}, \quad C(\mathbf{G}) = \mathbf{C} \tag{1}$$

Where the function C is symmetric:

$$C(X) = Y \iff C(Y) = X \quad \forall\, X, Y \in \{\mathbf{A}, \mathbf{T}, \mathbf{C}, \mathbf{G}\} \tag{2}$$

In the two helical chains, the complement of each nucleic acid is always at the other side in the other helical chain. This way, one DNA strand can copy itself by separating the two strands and adding the complement to both single helical strands, as shown in Fig. 1(b). Now, to give a representation to the DNA-strands we untangle the two helical strands and see the following structure in Fig. 2. Where each | represents a chemical bond between the nucleic acid above and below the chemical bond. In order to read the information that is stored in the DNA, we must know how living cells retrieve the information from the DNA. It appears that the DNA is read from the 5' (five prime) to the 3' (three prime). For our convenience we can read from left to right, placing the 5' at the left and 3' at the right. Because the two helical strands of DNA are anti-parallel and the information is stored in only one of the two helical strands, the strands in Fig. 2 codes for two different strands: ACTGATCGATC and GATCGATCAGT. The reason for this will be explained in Section 1.2.

Finally we could say that the language of DNA consist of the following

```
5'  ACTGATCGATC  3'
    | | | | | | | | | | |
3'  TGACTAGCTAG  5'
```

Figure 2: Representation of a DNA strand.

One-letter code	Three-letter code	Name
A	Ala	Alanine
C	Cys	Cysteine
D	Asp	Aspartic acid
E	Glu	Glutamic acid
F	Phe	Phenylalanine
G	Gly	Glycine
H	His	Histidine
I	Ile	Isoleucine
K	Lys	Lysine
L	Leu	Leucine
M	Met	Methionine
N	Asn	Asparagine
P	Pro	Proline
Q	Gln	Glutamine
R	Arg	Arginine
S	Ser	Serine
T	Thr	Threonine
V	Val	Valine
W	Trp	Tryptophan
Y	Tyr	Tyrosine

Table 1: All standard amino acids.

strings:

$$DNA = \{ \begin{matrix} A \\ | \\ T \end{matrix} , \begin{matrix} T \\ | \\ A \end{matrix} , \begin{matrix} G \\ | \\ C \end{matrix} , \begin{matrix} C \\ | \\ G \end{matrix} \}^{+} \tag{3}$$

1.2 Genes

Genes are the coding parts of DNA. Genes are segments of the genome and code for the active compounds inside a cell, called proteins. The proteins take care of all the biochemical activity in the cells of organisms. Proteins consist of a sequence of amino acids, ranging from several hundreds to thousands of amino acids, like the protein Titin, which consist of $27,000$ amino acids. Shortly after or even during synthesis, the residues in a protein are often chemically modified, which alters the physical and chemical properties, folding, stability, activity, and ultimately, the function of the proteins[4]. There are 20 different amino acids and they all have a three-letter and one-letter abbreviation, shown in Tab. 1. For convenience reasons, the representation of proteins will be shown using the one-letter abbreviation of the amino acids and the thus language of

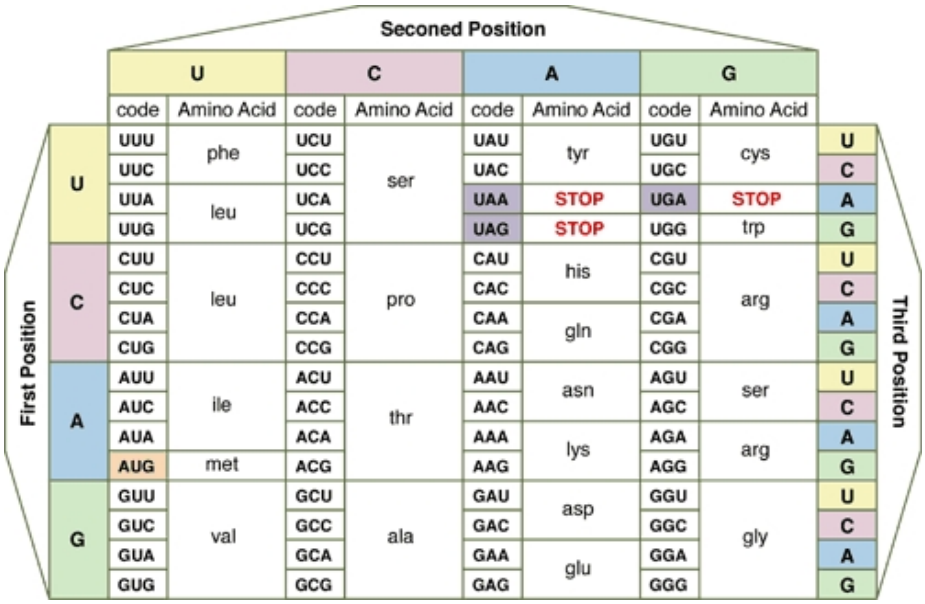


Figure 3: Codon table, translation from RNA to amino acids.

all proteins can be written as:

$$Proteins = \{A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y\}^+$$

The mechanism that is used to synthesize proteins from DNA is split up into 3 different stages: *Transcription*, *Translation* and *Splicing*[5]. Transcription is the process of making a single stranded RNA molecule, the messenger RNA (mRNA), from the coding DNA segment, the gene, whose sequence is complimentary to the DNA segment from which it was transcribed. RNA is almost equivalent to DNA and **T**hymine is replaced by **U**racil, a fifth nucleotide. The gene is always located on one of the two helical chains, which is called the coding strand and the template strand, the other strand, is used to make a copy of the coding strand.

Translation is the process of using the newly synthesized mRNA as a template for the synthesis of a new protein. The mRNA is read three nucleotides at a time, in units called codons. Each codon represents an amino acid, as shown in Fig. 3. Every newly synthesized protein must start with Methionine, the Start-codon, AUG. After this, three nucleotides at a time are read from the RNA strain, until one of the three stop codons appears, which is the end of the protein. The protein is now fully synthesized. An example of how transcription and translation operate, is shown in Fig. 1.2. All prokaryotes undergo this process of protein-making. With eukaryotes it is slightly different. The mRNA of eukaryotes is subject to a process called splicing. Therefore, after transcription, in eukaryotes, the product is pre-mRNA. Splicing removes non coding parts from the pre-mRNA, called introns, an leaves only coding parts, called exons, in the pre-mRNA, constructing mature mRNA or just mRNA, shown in Fig. 5. The mechanism that determines the introns and exons is not fully known, however the GT-AG rule is a good rule of thumb. The GT-AG rule is the obser-

DNA: ACGATGTGCGTACGATCGTAGGAC
TGCTAGACGCATGCTAGCATCCTG

⇓ *Transcription*

RNA: ACGAUGUGCGUACGAUCGUAGGAC

⇓ *Translation*

Resulting protein: M C V R S *

Figure 4: The process of synthesis of proteins by Transcription and Translation. The upper DNA strain is the coding strand for the new protein. Using transcription, the mRNA is constructed by creating a complement of the template DNA. The mRNA is then translated by looking at the first start codon and coding the protein until a stop codon is found. The underlined parts of the mRNA strain are the start and stopcodons. The asterisk denotes the end of the protein, but unlike the start amino acid, there is no stop amino acid.

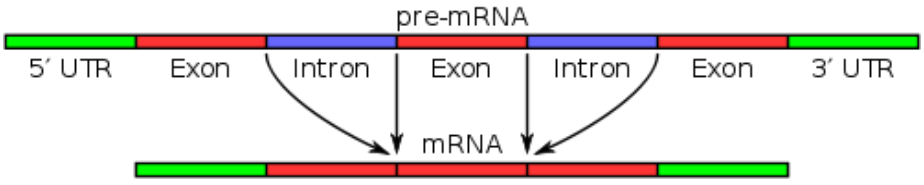


Figure 5: Simple illustration of exons and introns in pre-mRNA and the formation of mature mRNA by splicing. The UTRs are non-coding parts of exons at the ends of the mRNA.

vation that all introns in DNA begin with the nucleotides of GT and end with the nucleotides AG. When the DNA is transcribed into RNA, the introns are removed from the RNA by a mechanism that recognizes these beginning and ending nucleotides. In addition to the rule, the **AG** is sometimes replaced by AC.

1.3 Sequencing

In order to retrieve the information stored in the DNA we first must read it. As easy as this sounds, one must realize the size of DNA and the complexity of a living cell. The size of one nucleic acid is approximately 10 Å, far smaller than the scope of a optical microscope (which is 2000 Å). The 'reading' is therefore not doable by optical microscopy, but several other techniques are used. Two techniques will be discussed, the first being Chain Termination Method and the latter one being Illumina sequencing.

The Chain Termination Method was one of the first sequencing techniques and became fairly popular due to the fact that is required less toxic chemicals and lower amounts of radiation, and therefore became the method of choice in the '70. Although there are multiple Chain Termination Methods, they all are based on the same idea. The classical Chain termination Method requires a mixture of single stranded DNA templates all with the same sequene, a DNA primer which serves as a starting point, DNA polymerase to complete the single stranded DNA strands, normal nucleotides (A, T, C and G) and modified nucleotides. The modified nucleotides are modified in a way that the copying procedure of DNA, which is done nucleotide per nucleotide, will abort once this nucleotide is attached to the DNA template. This nucleotide is also labeled with a radioactive or fluorescent marker. When all the components are put together, the DNA polymerase will try to complete the single stranded DNA template, by adding the nucleotides (A, T, C and G) one-by-one. At a random point in time, a modified nucleotides will be added to the template strand, causing the chain reactions of adding nucleotides to terminate. This will results in a mixture of DNA strands with the same starting point, but different lengths. A process called Gel electrophoresis will then separate the strands based on length, by forcing the DNA strands to move through a gel. The smaller DNA fragments will move faster through the gel than the larger ones, separating the different DNA fragments. The DNA fragments are then visualized with X-ray, showing the fragments with a terminating modified radioactive nucleotide. Fig. 6 shows the result of the gel electrophoresis, with the smaller DNA fragments being at the bottom and the larger ones at the top. Now, the DNA strands can then be read by looking which band is the lowest and then going up until the highest one. The strand sequenced would have the following code: TACGAGATATAT...

Illumina sequencing, is one of the newer methods [6]. It uses reversible dye-terminators. The genome that is to be sequenced, is fragmented and placed on a slide. Four types of modified nucleotides are then added, and the non-incorporated nucleotides are washed away. The four modified nucleotides can only be added one-by-one and after the nucleotides are attached to the DNA strands, a camera takes images of the fluorescent labeled nucleotides. All the fluorescent terminating labels are then removed and the the four types of modified

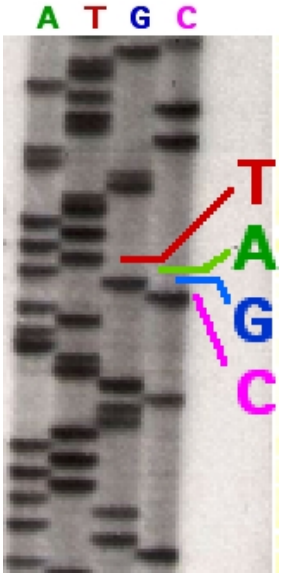


Figure 6: Part of a radioactively labeled sequencing gel

Sequence 1	A	T	C	T	G	G	A	T	C	G	A	T	
Sequence 2		T	C	T	G	G	A	T	C	G	A	T	A
Resulting sequence	A	T	C	T	G	G	A	T	C	G	A	T	A

Figure 7: The process of a sequence assembler, comparing two overlapping sequences

nucleotides are added again for the next cycle.

After the sequencing of the genome of a certain organism, the output is a huge number of strings containing the 4 different nucleotides with an average length of 26-60. Using a sequence assembler, a computer program that checks for overlap between two different strings, the length of these strings can be extended, shown in Fig. 7.

Using this technique the length of strings can increase dramatically, up to thousands of nucleotides per string.

1.4 Blast

When analyzing a genome, comparing sequences is a very effective approach. For instance, if we want to know if a certain gene is part of a genome, we look for a sequence that is fairly similar to the gene we are looking for. This similarity is called homology. Homology originates from having a common ancestor. For example, if two or more genes have highly similar DNA sequences, it is likely that they are homologous. But sequence similarity may also arise without common ancestry: short sequences may be similar by chance, and sequences may be similar because both were selected to bind to a particular protein, such as a transcription factor. Such sequences are similar but not homologous. Sequence regions that are homologous are also called conserved. This similarity property is

Algorithm	Description
blastn	Compares DNA-sequences with a DNA database
blastp	Compares protein sequences with a protein database
blastx	Translates a DNA sequence into all six reading frames, and compares these translations with a protein database.
tblastn	Compares a protein sequence with all six reading frames of the DNA sequences in a database
psi-blast	This algorithm is used to find distant relatives of a protein. First, a list of all closely related protein is constructed and a profile of these sequence is made. This profile is then used to compare the profiles of sequences in the database
tblastx	Translates a DNA sequence in all six reading frames and compares this with the translated DNA sequences of a DNA sequence database, in all six reading frames.

Table 2: Different BLAST algorithms with their description.

used by comparing programs that compare sequences of DNA, RNA and amino acids. The most common tool to compare sequences is *BLAST*. BLAST, short for Basic Local Alignment Search Tool, is an algorithm for comparing primary biological sequence information, such as the amino-acid sequences of different proteins or the nucleotides of DNA sequences [7]. BLAST is the most commonly used comparing program for Bioinformatics studies. The program is very elaborate and every search can be tuned to get the best results. There are 6 main BLAST algorithms: *blastn*, *blastp*, *tblastn*, *blastx*, *psi-blast* and *tblastx*. Tab. 2 enumerates and explains the main purpose of these algorithms. 2 of these 6 algorithms use a translated sequence, by translating over all the possible reading frames. An mRNA has three reading frames, each reading frame corresponds to starting at a different alignment to read the three lettered codons. Double stranded DNA has 6 reading frames, due to the two strands from which transcription is possible: three of them reading forward and three of them reading backwards, as shown in Fig. 8.

1.5 Blast2GO

In Bioinformatics there is a lot of information to grasp for research. Especially with sequence databases, this amount of information can be overwhelming. Furthermore, because the wide variations in terminology, finding a specific subset of genes can be difficult. For instance, if you were looking for new targets for antibiotics, a good start would be to assemble all the genes that are involved in the protein synthesis. If one database would describe the molecules you're looking for as 'translational proteins' and in another it would refer to 'protein synthesis proteins', the task to gather this subset of molecules would be difficult for both you and a computer.

To establish more consistency, the Gene Ontology (GO) project was started. The GO project has developed three structured controlled vocabularies (ontologies) that describe gene products in terms of their associated biological processes, cellular components and molecular functions in a species-independent manner [8]. The ontology trees can be consulted at multiple levels. For exam-

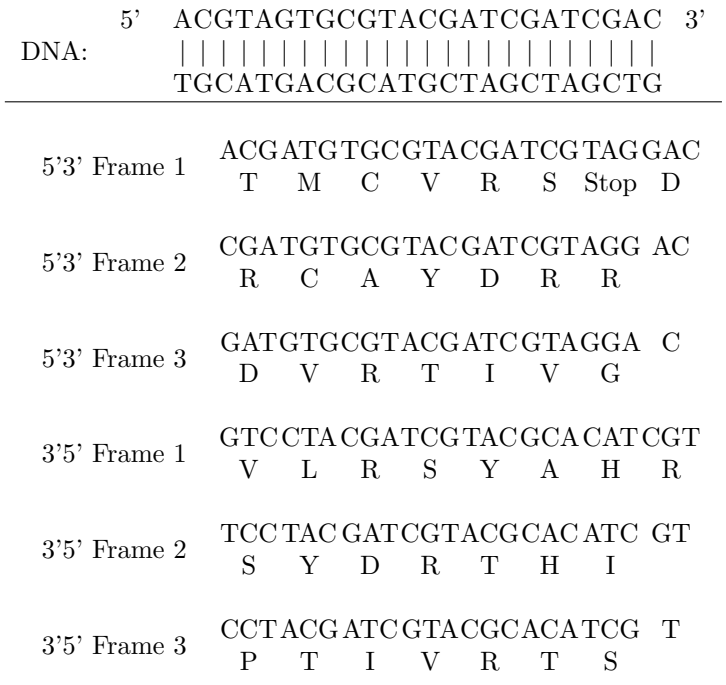


Figure 8: The 6 reading frames of a double stranded DNA strain.

ple, you can use GO to look for all the signal transduction proteins of a certain genome, or zoom in and find all the receptor tyrosine kinases.

Blast2GO is a research tool designed with the main purpose of enabling GO based data mining on sequence data for which no GO annotation is yet available [9]. B2G combines GO annotation based on similarity searches with statistical analysis and highlighted visualization on directed acyclic graphs. Blast2GO generates combined graphs where the combined annotation of a group of sequences is visualized together. This can be used to study the joined biological meaning of a set of sequences. These combined graphs are a good alternative to enrichment analysis where there is no reference set to be considered or the number of involved sequences is low.

The program uses three steps: *BLAST*, *Mapping* and *Annotation*. The sequences, either DNA or protein, that are loaded into the program, are the input for a BLAST run. This information is then saved for the next step: Mapping. Mapping is the process of finding GO terms related to the BLAST results. This is done in two ways. When BLAST recognizes similarity between a query sequence and a database sequence, two identifiers are saved: the sequence accession and the GenInfo (GI) identifier. The accession can be used to search the databases of the National Center for Biotechnological Information (NCBI) for GO terms. The GI identifier can be used to search other databases for GO terms. After every sequence has GO terms, the Annotation step is next. All the GO terms that were assembled in the mapping phase, are now filtered by applying an annotation rule (AR) on the found GO term, to the extend of seeking the most specific annotations with a certain level of reliability. The annotation rule is

composed of two terms:

$$AR = DT + AT \quad (4)$$
$$DT = max\_similarity \times EC_{weight} \quad (5)$$
$$AT = (\#GO - 1) \times GO_{weight} \quad (6)$$

where $max\_similarity$ is the maximum value of similarity between the query and hit sequences that have the given GO annotation, EC_{weight} the weight given to the Evidence Code of the original annotation $\#GO$ is the number of GO terms associated to that query and GO_{weight} is the weight given to the contribution of annotated children term to a given term. The annotation rule provides a general framework for annotation. The actual way annotation occurs depends on how the different parameters at the AS are set.

1.6 Definitions of Problems

For this research project, the genome of *L. Stagnalis* was sequenced. This was done using the Illumina technique and the output of this sequencing is the assembly. With this information, we formulated the following questions for our research:

1. With the data we have, is the process of annotation a possibility?
   (a) Is there a set of conserved genes?
      i. Can we find these conserved genes in our genome?
      ii. Are there hox genes of other species that we can find in our genome?
2. What can Blast2GO tell us about the raw data?
   (a) Does pre-screening assist the Blast2GO annotation process?

2 Materials and Methods

2.1 Materials

The assembly contains over 700 Megabytes of information of DNA. The coverage of the assembly is 1.69. Coverage is the average number of reads representing a given nucleotide in the reconstructed sequence. The entire assembly is divided over 681705 contigs, uninterrupted parts of the genome, ranging from 100 nucleotides to maximum 51000 nucleotides, with the median being at 2417 nucleotides. All contigs are put together in one file, in the *FASTA*-format, which has the following composition:

```
>contigname
CACTCAGTACGATCAGCTAGCATCGATCAGCTACGACTAGCTGAGCTGACTGAGCGATCGATCG
AGCATGTACGATCGATCGACTAGCTAGCTACGATCGCTAGCTACGATCG
>next_contigname
ATGCATGATCGATCGAGCTAGCTAGCATCGACNNNNNNNNNNNNAGACTAGTAGC
```

Each contig begins with a " >″ followed by some information about that contig. This can either be the name of that contig, the GO-id or combinations of these tags. After the linefeed the sequence of that part of the genome begins with the notation of a single stranded DNA strain. Because the complement of DNA is deterministic, this is a good way of information compression. To illustrate, the sequence of `contigname` is the following:

```
5' CACTCAGTACGATCAGCTAGCATCGATCAGCTACGACTAG···
   |||||||||||||||||||||||||||||||||||||||||||
3' GTGAGTCATGCTAGTCGATCGTAGCTAGTCGATGCTGATC···

···CTGAGCTGACTGAGCGATCGATCGAGCATGTACGATCGAT···
   |||||||||||||||||||||||||||||||||||||||||||
···GACTCGACTGACTCGCTAGCTAGCTCGTACATGCTAGCTA···

···CGACTAGCTAGCTACGATCGCTAGCTACGATCG  3'
   |||||||||||||||||||||||||||||||||
···GCTGATCGATCGATGCTAGCGATCGATGCTAGC  5'
```

Notice the N in `next_contigname`, which represents an unknown nucleotide, and together they form a gap. However, they give some extra information, because we now know what the distance is between the two strains.

To tackle the first problem, whether our assembly is reliable enough for annotation, we're going to check if there is a set of genes that is very conserved within the genomes of many species. This will show if the assembly we have, is consistent with the normal findings of a genomes analysis. The hox gene is an example of a conserved gene. Hox genes are a group of related genes that determine the basic structure and orientation of an organism. Hox genes are a subset of homeobox genes: genes containing the homeodomain in their translated sequence. The homeodomain is a sequence of 60 amino acids, that binds to DNA. Remarkable is that, unlike other sorts of genes, their sequence has variated very little. Furthermore, hox genes are very similar to hox genes in near by species. In extend to their similarity, their organization is in chromosomal clusters, consisting of 8 genes in the fruit fly, *Melanogaster Drosophila*, and 39 Hox genes in 4 complexes in mammals. Also, a spatial collinearity is seen with hox genes. The gene order in the cluster mimics the order of expression of genes and their function along the anterior–posterior (A–P) body axis: genes at the 5' end of the cluster are expressed in, and pattern, the posterior part of the body, whereas genes at the 3' end pattern the anterior end of the body [10]. The consensus of the homeodomain is shown in Fig. 9. One convenient property of hox genes is they consist of two exons and one intron. If two parts of the hox gene are found on two different contigs, all the information is than available to reconstruct the sequence of the hox gene. To check if the hox genes are present in the assembly, we used a 8 core computer with Linux and blast+ v2.2.25. To look for the hox genes, the National Center for Bioinformatics Information (NCBI) website[11] is a very good feed for any kind of Bioinformatics information. Both sequences and the program blast (on-line and off-line version) is supported by this institute. The off-line version was installed on the Linux computer. The on-line version was used for quick inquiry during the project. For the second

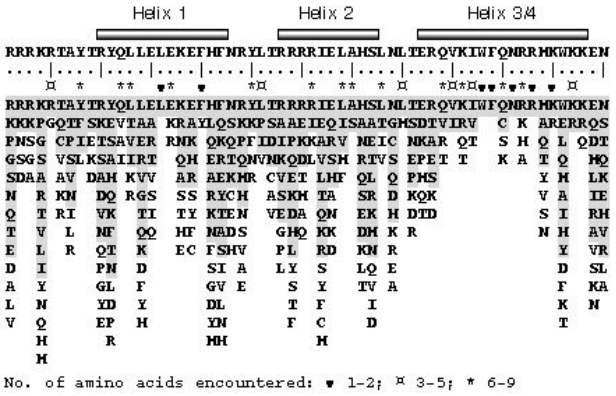


Figure 9: The consensus of the homeodomain, with the most frequent amino acids at the top.

question, what kind of information will Blast2GO provide us, we introduced a way to reduce the amount of data to analyze, because of the access of data. Blast2GO uses blast results to identify and combining the information stored in all the data bases, explained in Section 1.5. Because of the huge amount of sequences, a subset was created based on the lengths of the open reading frame. An open reading frame (ORF) is a sequence of nucleotides which begins with a start codon and ends with a stop codon. There are 6 reading frames for every DNA strain, three in each direction. If we look at a certain DNA sequence and know it is coding, e.g. a gene, the length of the open reading frame will be significantly longer in one of these 6 reading frames. If this manner is used, we can see whether pre-screening, in our case creating a subset based on the ORF, will effect the outcome of the Blast2GO. For this, we have a local GO database installed on a Linux computer, with 8 cores.

2.2 Methods

2.3 Lymnaea Stagnalis Hox Genes

To search for hox genes, the NCBI site was resourceful. The protein database that was used, is a collection of sequences from several sources, including translations from annotated coding regions in GenBank, RefSeq and TPA, as well as records from SwissProt, PIR, PRF, and PDB [12]. This protein database supported us with 13 sequences: 11 partial hox genes and a partial para hox gene and a partial NK gene. Both the para hox and the NK gene have the same common ancestor cluster as hox genes: at some point in time a general cluster called the *ProtoANTP* cluster divided up into two different clusters, the Protohox cluster and the NK cluster. The Protohox cluster was then split up into the Hox cluster, and the ParaHox cluster. Table 3 shows the hox(like) genes that were retrieved from the NCBI. Using the blast+ program, the partial hox gene were searched for in the assembly.

Name	Type	Partial Sequence
Msx-like HP	NK	REKQYLS I AER A EF S A S LTLTETQVK I
Xlox-like HP	ParaHox	HFNKY I S RPRR I E L AAML S LTERH I
Post2 HP P2	Hox	LNS S Y I TRQKRWE I S CKLQL S ERQVKV
Post1 HP P1 allele2	Hox	A S S TY I S K S RRWE L S Q L I NL S ERQ I K I
Post1 HP P1 allele1	Hox	A S S TY I S K S RRWE L S Q L I NL S ERQ I K I
abdA-like HP M8	Hox	QFNHYLTRKRR I E I AH S LCLTERQ I
Ubx-like HP M7	Hox	KFNRYLTRRRR I E L S HMLCLTERQ I
Antennapedia-like HP M6b	Hox	HFNRYLTRRRR I E I AHMLGLTERQ I
Antennapedia-like HP M6a	Hox	HYNRYLTRRRR I E I AH S LALTERQ I
Scr-like HP M5	Hox	HYNKYLTRRRR I E I AH A LNLTERQ I
Hox3 HP	Hox	HFNRYLCRPRR I EMAA L LNLTERQ I
Hox2 HP	Hox	HFNKYLCRPRR I E I AA S LDLTERQV
Hox1 HP	Hox	HFNKYLTRARR I E I AA A LGLNETQV

Table 3: The hox(like) genes that were retrieved from the NCBI protein database, with their sequences. HP is an abbreviation for Homeodomain Protein. The genes Post1 HP P1 allele 1 and 2 have the same sequence, however the translated DNA is different (not shown). These partial sequences are all part of the homeodomain.

2.4 Homologs of Hox Genes in Lymnaea Stagnalis

Because NCBI provided only partial sequences of the hox genes, the remainder of the hox genes must be retrieved using a different manner. Using the homologs is a convenient way to search for certain genes. Two sets of hox genes were selected: hox genes of the fruit fly, *Melanogaster Drosophila* and *Caenorhabditis Elegans*. The genomes of these two species have been sequences multiple times and thus much is known about them. *M.drosophila* isn't close to the pool snail in the tree of life, but much is known about its hox genes. *C.elegans* is somewhat closer to the pool and thus might show more similarity. All the hox genes of these organisms, 8 for *M.drosophila* and 6 for *C.elegans*, were retrieved from several protein databases and blast+ was used to search for the similarity between the translated DNA of Lymnaea Stagnalis and the hox genes.

A second technique was to look for closely related species of the pool snail and look for their hox genes. This was done by supplying the on-line version of blast with the contigs (Tab. 5) that were similar to the hox genes of Tab. 3. The hox genes of the species with the best hits, were retrieved from the NCBI website and blasted against, to search for similar sequences.

A third technique was to search for specific hox genes in the protein database of the NCBI website. The query that was used to search for the hox genes was the following: `("hox_name"[Gene Name]) AND 61:2000[Sequence Length]`, where `hox_name` was replaced by a hox gene name. The Sequence length was chosen to be 61:2000 to search for proteins in the database that are longer than the homeodomain, a region that is very abundant in the database, but is not useful to us, since we're looking for the regions beyond the homeodomain. The following hox gene names were used to search for hox genes in our assembly: hox1, Post1, Post2, Scr, antp and ubx.

2.5 Blast2GO & ORF Reader

In order to analyze the data using Blast2GO and save time, a local BLAST run was performed. This was done using the SwissProt database, a manually annotated protein database, which makes a reliable source for proteins. Our assembly appeared to be too big to run a blast with, so a program was written to reduce the data, based on ORF length. The algorithm is explained in pseudo-code in Alg. 1.

Three datasets of DNA sequences were created this way, shown in Table 4. Be-

Algorithm 1 Compute the ORF length of a DNA sequence and include or exclude it in the subset that is created, based on the length.

```
Specify the length of the ORF's, L
Open the assembly file
for all Sequences, s, in the assembly do
  Nmax = 0
  for all Reading frames do
    Start ← true
    N ← 0 {N is the number of nucleotides in the ORF}
    for all Codons in this reading frame do
      if −Start and Codon = Start-codon then
        N ← 0
        Start ← true
      else if Start and codon = Stop-codon then
        if N ≥ Nmax then
          Nmax ← N
        end if
        Start = false
      end if
      N ← N +3
    end for
  end for
  if Nmax ≥ L then
    S ← S ∪ s
  else
    discard s
  end if
end for
return S
```

cause BLAST results are very elaborate, and therefore big in size, the datasets had to be split up into pieces, so it would be possible for Blast2GO to parse the BLAST output. With the use of *fastasplitn* [13], the datasets were split up into pieces of reasonable size ($<$ 300KB) so the output of BLAST wouldn't be too big ($< 15MB$). Using the following bash script, the three datasets were processed by BLAST and Blast2GO:

```
#!/bin/bash
FILES1="/u01/home/jneutebo/Bachelorproject/sequencedata/datasets/l1000"
FILES2="/u01/home/jneutebo/Bachelorproject/sequencedata/datasets/l750"
```

Name	ORF length in nucleotides	Size in MB	#Annotated sequences	Divided into #pieces
l1000	1000	30.9	1883	100
l750	750	45.2	2651	150
l500	500	91.3	4423	300

Table 4: The datasets that were created using Alg. 1, with name, ORF length, size and the number of parts it was split up.

```
FILES3="/u01/home/jneutebo/Bachelorproject/sequencedata/datasets/l500"
blast2go="/u01/home/jneutebo/Bachelorproject/blast2go/b2g4pipe"
properties="/u01/home/jneutebo/Bachelorproject/blast2go/b2g4pipe/b2gPipe.properties"
for f in "$FILES1"/*
do
  ./blastx -query $f -db /u01/home/jneutebo/Bachelorproject/blast/swiss1 -show_gis -evalue 1e-10 -num_threads
8 -outfmt 5 > $f.xml
  java -Xms4000m -Xmx20000m -jar $blast2go/blast2go.jar -in $f.xml -prop $properties -v -a -out $f.annot -d
done

for f in "$FILES2"/*
do
  ./blastx -query $f -db /u01/home/jneutebo/Bachelorproject/blast/swiss1 -show_gis -evalue 1e-10 -num_threads
8 -outfmt 5 > $f.xml
  java -Xms4000m -Xmx20000m -jar $blast2go/blast2go.jar -in $f.xml -prop $properties -v -a -out $f.annot -d
done

for f in "$FILES3"/*
do
  ./blastx -query $f -db /u01/home/jneutebo/Bachelorproject/blast/swiss1 -show_gis -evalue 1e-10 -num_threads
8 -outfmt 5 > $f.xml
  java -Xms4000m -Xmx20000m -jar $blast2go/blast2go.jar -in $f.xml -prop $properties -v -a -out $f.annot -d
done
```

Here, blastx was used to search for similarity between the datasets, `-query`, and the SwissProt database, `-db`. `-show_gis`, tells blastx to include the GenInfo of every sequence in the output, `-evalue` filters only the most significant results having an e-value of at least $1 \exp -10$, `-num_threads` is the number of processors that is used and `-outfmt 5` tells blast to format the output in an XML structure. The created XML-file is than input for the program `blast2go.jar` which is run by the Java virtual machine. It creates annot-files with query sequence names and GO-id's. Per dataset, all the annot files were reassembled and analyzed by Blast2GO. Blast2GO was then used to generate combined graphs, to visualize the combined annotation. For each of the three datasets that were created, the three main GO categories that Blast2GO offers (biological process, cellular functions and cellular component) were selected as the tree types and pie graph was constructed for comparison with the other datasets.

3 Results

3.1 Lymnaea Stagnalis hox genes

The hox genes of L.Stagnalis that were found at the NCBI, were used as input for a BLAST against the assembly. Results are displayed in Tab. 5.

Name	Contig number	Hit percentage	E-value
Msx-like HP	422981	100% (27/27)	1 exp −07
Xlox-like HP	136346	100% (25/25)	3 exp −07
Post2 HP P2	130330	100% (27/27)	2 exp −06
Post1 HP P1 allele2	138537	100% (27/27)	4 exp −07
Post1 HP P1 allele1	138537	100% (27/27)	4 exp −07
abdA-like HP M8	407902	100% (25/25)	3 exp −07
Ubx-like HP M7	407695	100% (25/25)	4 exp −07
Antennapedia-like HP M6b	636650	100% (25/25)	6 exp −07
Antennapedia-like HP M6a	417153	100% (25/25)	3 exp −07
Scr-like HP M5	161502	100% (25/25)	8 exp −07
Hox3 HP	144188	100% (25/25)	8 exp −09
Hox2 HP	630928	100% (25/25)	7 exp −08
Hox1 HP	139291	100% (25/25)	5 exp −07

Table 5: BLAST result of the L.Stagnalis hox genes against the assembly. Columns are the name of the hox gene from NCBI, the number of the contig from the assembly, the score of the alignment made by BLAST and the E-value.

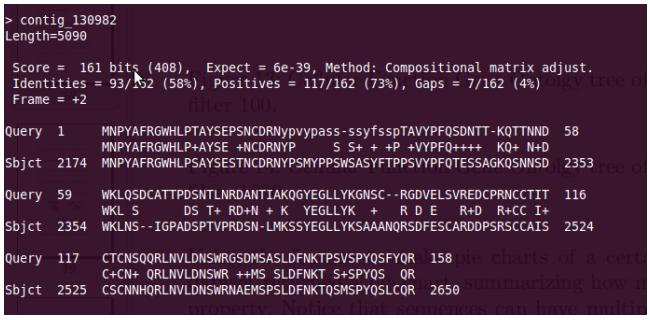


Figure 10: Blast results of the comparison between contig_130982 and Gva post2.

3.2 Hox gene homologs of Lymnaea Stagnalis

The approaches used to further annotate the hox genes resulted two findings. Using the second approach, by looking at hox genes of relatives which where proposed by the on-line version of BLAST, two species showed similarity with their hox genes: Posterior hox gene of Gibbula varia, Gva post2, equal to Post1 HP P1 and Hox5 of Haliotis Rufescens similar to Scr-like HP M5. Alignment will be shown in Fig. 10 and Fig. 11 for Gva post2 and Hox5, respectively. Using the same approach, contig 208312, showed some similarity. When the on-line blast was used to analyze this contig, it showed remarkable similarity with hundreds of hox4 genes of different species. What's interesting is that the contig seems to contain two parts of the homeodomain, separated by an exon of 1166 nucleotides long.

3.3 Blast2GO & ORF reader

Three datasets where used to create three tree of every GO tree type.

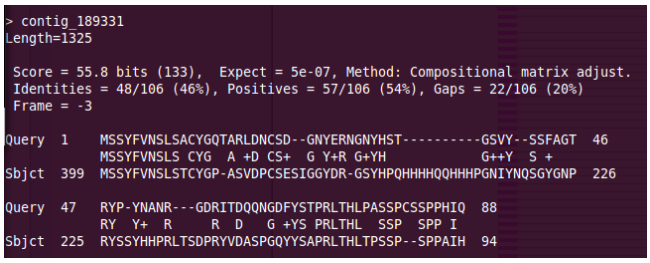


Figure 11: Blast results of the comparison between contig_189331 and Hox5.

3.4 Blast2GO Tree Figures

3.4.1 *l1000*

Using a filter to exclude insignificant nodes from the tree, three ontology trees were created with the *l1000* dataset. Biological process, cellular function and cellular component are shown in Fig. 15, Fig. 16 and Fig. 17, respectively, in the Appendix.

3.4.2 *l750*

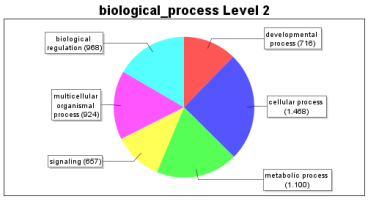
Using a filter to exclude insignificant nodes from the tree, three ontology trees were created with the *l750* dataset. Biological process, cellular function and cellular component are shown in Fig. 18, Fig. 19 and Fig. 20, respectively, in the Appendix.

3.4.3 *l500*

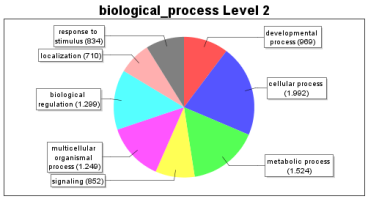
Using a filter to exclude insignificant nodes from the tree, three ontology trees were created with the *l500* dataset. Biological process, cellular function and cellular component are shown in Fig. 21, Fig. 22 and Fig. 23, respectively, in the Appendix.

3.5 Dataset Comparison

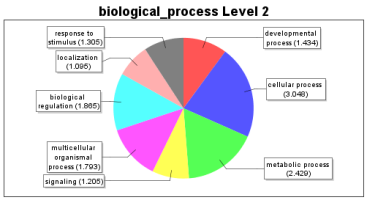
To compare the three datasets, pie charts were created for every GO tree, to compare the number of sequence per node at the second level of the trees. Using the feature to make pie charts of a certain level of a GO tree, every dataset returned a pie chart, summarizing how many sequences have a specific property. Notice that sequences can have multiple properties and thus belong to more than one category of the same level. All of the pie charts represent the number of sequence in a certain category, at the second level, the first being the root. The Biological Process summary of *l1000*, *l750* and *l500* are shown in Fig. 12(a), 12(b), 12(c), respectively. The Molecular Functions summaries of *l1000*, *l750* and *l500* are shown in Fig. 13(a), 13(b), 13(c), respectively. The Cellular Component summaries of *l1000*, *l750* and *l500* are shown in Fig. 14(a), 14(b), 14(c), respectively.



(a) *l1000*

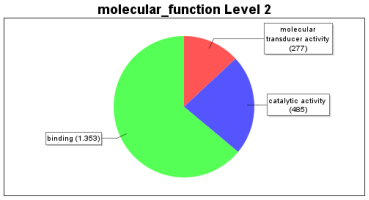


(b) *l750*

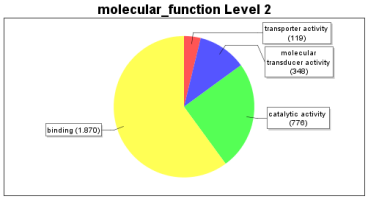


(c) *l500*

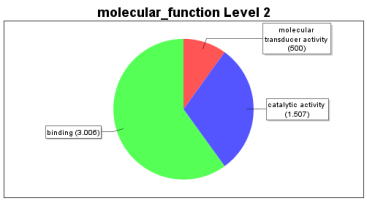
Figure 12: Summary of the Biological Process GO trees, for all three datasets.



(a) *l1000*

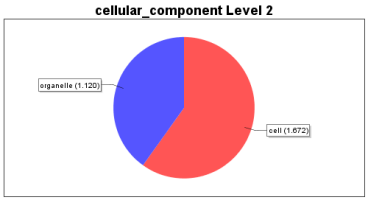


(b) *l750*

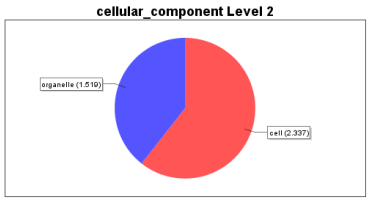


(c) *l500*

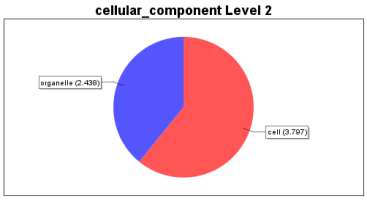
Figure 13: Summary of the Molecular Functions GO trees, for all three datasets.



(a) *l1000*



(b) *l1000*



(c) *l1000*

Figure 14: Summary of the Cellular Component GO trees, for all three datasets.

4 Discussion & Conclusion

To conclude this project, the definition of problems will be discussed.

Is there a conserved set of genes and can we locate them in our assembly? Taking hox genes as our conserved set of genes, we can definitely retrieve these partial genes. As the results show, all the already annotated parts of the hox genes, retrieved from the NCBI website, are located with a 100% accuracy.

Are there hox genes of other species that we can find in our genome? Using different approaches, finding hox genes in our assembly appeared to be harder than expected. Although the homeodomain was easily retrieved from the hoxgenes, the remaining parts of these genes is still the difficult part. Using the hox genes of *C.elegans* and *D.melanogaster*, no significant results were obtained, other than the similarity between the homeodomains. Utilizing hox genes of related species appeared to have more success, however the desired result to annotate all the hox genes was not reachable with our data. The hox gene Gva post1 was useful to locate the remainder of posterior homeobox protein P1 *L.stagnalis*. Knowing that hoxgenes consist of two exons, the sequence of posterior homeobox protein P1 could be determined in its entirety, see Appendix, because it was similar to Gva post1 from amino acids 1 to 158 and on an other contig from amino acid 159 to 227, in which the homeodomain is located. If we search for the stop codon in the reading frame of the last part, the entire gene is annotated. With the annotation of Scr-lie HP M5, the two parts we had were amino acid 1 to 88 for the beginning of the protein and 195 and to 255 being the homeodomain. Because these aren't neighboring, we can't say with 100% accuracy how the gene is build, because we don't know where the intron is positioned. Therefore, only two partial annotation were made. The first part is shown in the Appendix.

The discovery of part of hox gene hox4 in the assembly was an unexpected

surprise. The sequence of contig 208312 was used as input for the on-line version of blast, to investigate an uncommon similarity, which was found when comparing relative species against the assembly. The on-line version shows an alignment of the first part of the homeodomain, than a 1166 nucleotides long intermediate and than the rest of the homeodomain. This intermediate string follows the gt-ac rule and thus we conclude this is an exon, however not in the same reading frame. If we follow the rule of hox genes having only one exon, we should find a start codon somewhere in front of the first homeodomain. Instead a stop-codon is found and with the information we have, no entire annotation of hox4 can be done. This might be because the theory that hox genes have only one hox gene, is not 100% accurate. A probable reason to why this gene was not published by the authors that published the other Lymnaea stagnalis hox genes, is that they looked for the hox genes using a 27 long amino acid sequence. The hox4 gene is split up into two introns in this region.

With the data we have, is the process of annotation a possibility?
The first question in our research was, if we were dealing with an accurate assembly. To answer this question, we based it on the annotation of part of the hox genes we retrieved from the NCBI website. Based on the result of finding all the retrieved hox genes and that we found a new one we can say that the quality of the assembly is good. Unfortunately, we cannot say something about the completeness of the assembly.

What can Blast2GO tell us about the raw data?
If we look at the largest data set which consists of 23839 sequences, only 4423 sequences were annotated and mapped to GO-ids. That means that 81% of the assembly is not coding DNA based on the ORF theory, using a E-value filter of $1 \exp -10$. If we look to the tree created by Blast2GO, it is hard to say anything of significance. If we look at the pie graphs of the data set *l500*, the DNA with ORF's of minimal 500 nucleotides codes for proteins which are for 61% acting in the cell, whereas the remainder acts in organelles. Looking at the Molecular Function, 60% of the GO-id map to binding proteins, 10% to molecular transducer activity and 30% to catalytic activity. When looking at Biological Process, the main three categories the GO-ids map to are cellular processes, (22%), metabolic processes, (17%) and biological regulation (13%). This however does not say that, for instance, 22% of the coding DNA codes for cellular process, because every sequence has at least one GO-id and it can go up to 70 GO-ids, which might be confusing.

Does pre-screening assist the Blast2GO annotation process?
In this project, we have suggested the ORF length as a property to screen for. The pre-screening process therefore was creating the three data sets, *l1000*, *l750* and *500*, with the goal of reducing the data and thus reducing the time to analyze assemblies. If we look for radical differences, nothing is spotted in the cellular component pie graphs and we therefore conclude that pre-screen does not effect the cellular component sequences. When we have a look at the Molecular Function pie graphs, a slight change in the *l750* graph is seen, because a fourth category is joined, which is left out in the *l500* data set. This is because is takes much time creating these graphs and adjusting the sequence filter to the right value could take days. However we see comparable figures. In the Biological

Process graphs, we see no change except for the two joined categories, which has the same cause as with the Molecular Function. We conclude therefore that these three main GO categories are not effected by the ORF screening. However, stating these conclusions it must be taken in to account that we didn't have a comparison with the entire assembly as dataset.

References

[1] Wikipedia.org (15 June 2011), Bioinformatics - Wikipedia the free encyclopedia, *http://en.wikipedia.org/wiki/Bioinformatic*

[2] Wikipedia.org (20 July 2011), Genome - Wikipedia the free encyclopedia, *http://en.wikipedia.org/wiki/Genome*

[3] Wikipedia.org (28 June 2011), DNA - Wikipedia the free encyclopedia, *http://en.wikipedia.org/wiki/DNA*

[4] Wikipedia.org (23 July 2011), Protein - Wikipedia the free encyclopedia, *http://en.wikipedia.org/wiki/Protein*

[5] Wikipedia.org (23 July 2011), Gene - Wikipedia the free encyclopedia, *http://en.wikipedia.org/wiki/Gene*

[6] Next-Generation DNA Sequencing Methods, Annual Review of Genomics and Human Genetics Vol. 9: 387-402

[7] Wikipedia.org (23 July 2011), BLAST - Wikipedia the free encyclopedia, *http://en.wikipedia.org/wiki/BLAST*

[8] An Introduction to the Gene Ontology, *http://www.geneontology.org/GO.doc.shtml*

[9] Conesa, A., Goetz, S., García-Gómez, J. M., Terol, J., Talón, M., and Robles, M. (2005). Blast2go: a universal tool for annotation, visualization and analysis in functional genomics research. *Bioinformatics, 21(18):3674-3676.*

[10] Jordi Garcia-Fernández, The Genesis and Evolution of Homeobox Gene Clusters, *Nature Reviews Genetics*, volume 6, p881-892.

[11] *http://www.ncbi.nlm.nih.gov/*

[12] *http://www.ncbi.nlm.nih.gov/protein*

[13] *ftp://saf.bio.caltech.edu/pub/software/molbio/fastasplitn.c*

5 Appendix

5.1 Hox gene annotations

Reconstructed sequence of Posterior homeodomain protein P1 and the homeodomain of hox4, in FASTA-format.

```
>Post1_homeobox_protein_P1|Lymnaea_Stagnalis
MNPYAFRGWHLPSAYSESTNCDRNYPSMYPPSWSASYFTPPSVYPFQTESSAGKQSNNSD
WKLNSIGPADSPTVPRDSNLMKSSYEGLLYKSAAANQRSDFESCARDDPSRSCCAISCSC
NNHQRLNVLDNSWRNAEMSPSLDFNKTQSMSPYQSLCQREMQQSIHLPSTSIAPTTVTLR
KRRRPYSKFQIAELEREYASSTYISKSRRWELSQLINLSERQIKIWFQTRRIKAKKLQKR
EDMGVPQSSSNMGTPQPPQHLIQSSISVS

>Hox4_homeodomain|Lymnaea_Stagnalis
PKRARTAYTRHQILELEKEFHFNRYLTRRRRIEIAHTLDLSERQIKIWFQNRRMKWKKEH

>Scr-like_HP_M5|Lymnaea_Stagnalis
MSSYFVNSLSTCYGPASVDPCSESIGGYDRGSYHPQHHHHQQHHHPGNIYNQSGYGNPRY
SSYHHPRLTSDPRYVDASPGQYYSAPRLTHLTPSSPSPPAIH
```

5.2 Gene Ontology Trees

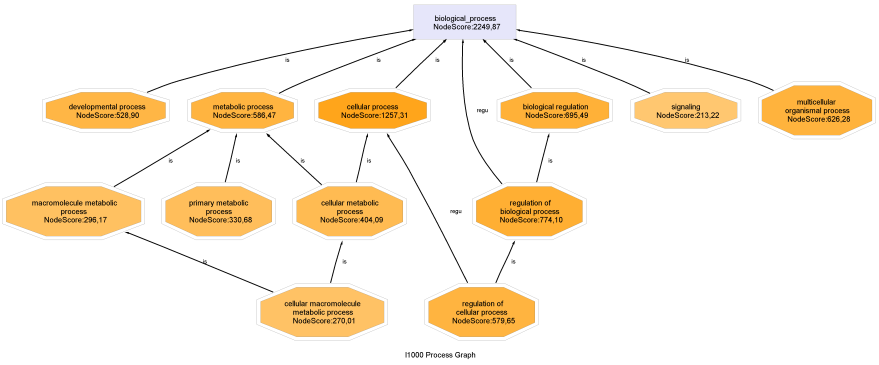


Figure 15: Biological Process Gene Ontology tree of dataset *l1000*, using sequence filter 600.

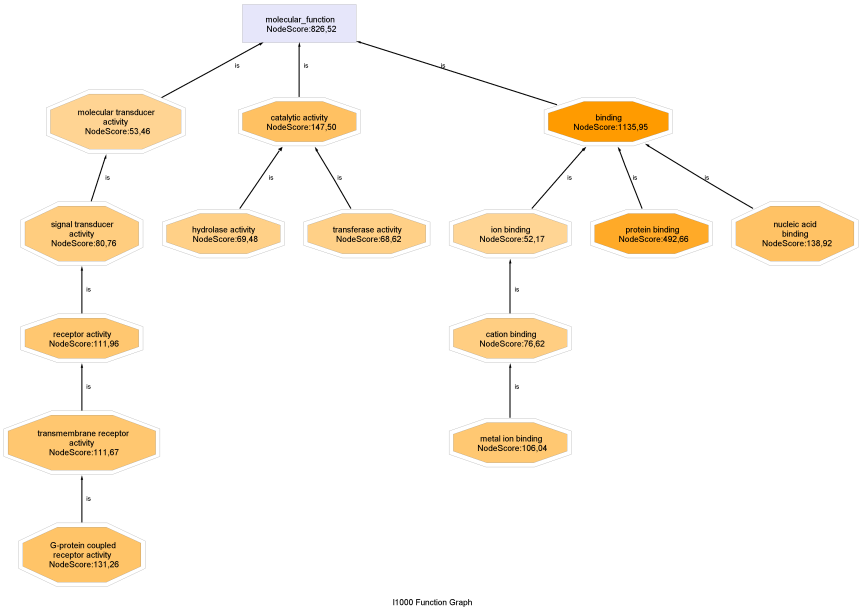


Figure 16: Molecular Function Gene Ontology tree of dataset *l1000*, using sequence filter 100.

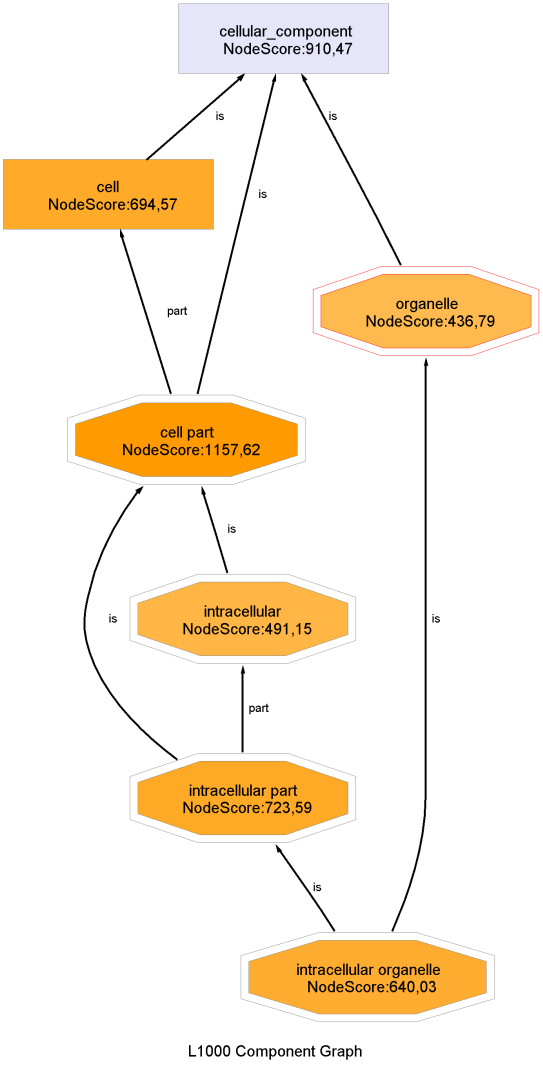


Figure 17: Cellular Component Gene Ontology tree of dataset *l1000*, using sequence filter 1000.

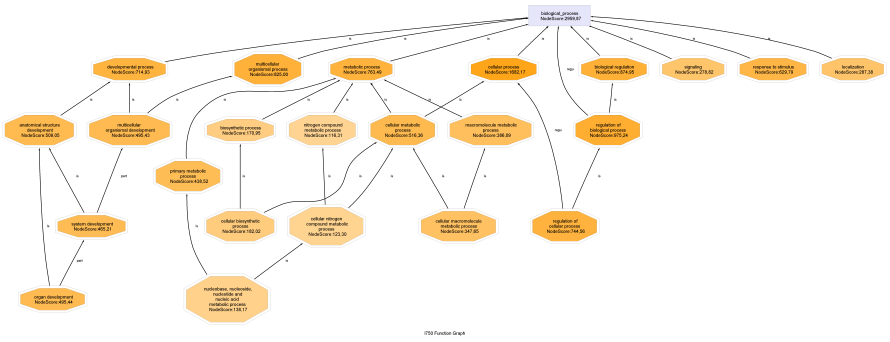


Figure 18: Biological Process Gene Ontology tree of dataset *l750*, using sequence filter 600.

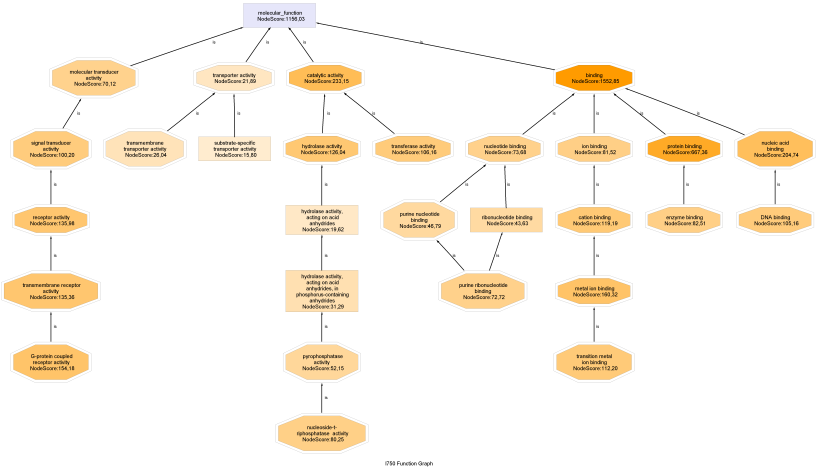


Figure 19: Molecular Function Gene Ontology tree of dataset *l750*, using sequence filter 150.

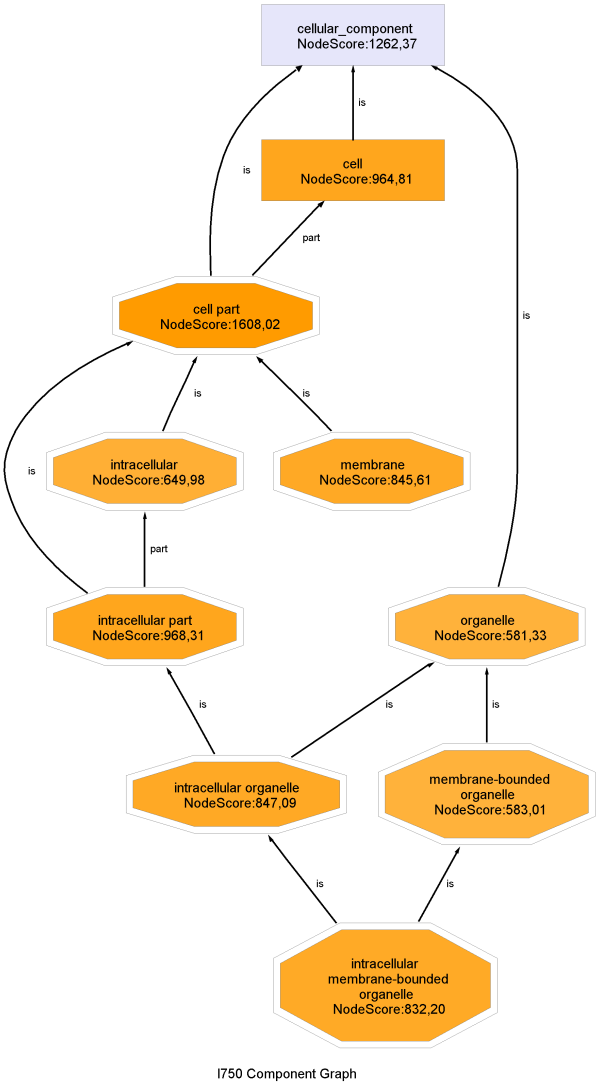


Figure 20: Cellular Component Gene Ontology tree of dataset *l750*, using sequence filter 1000.

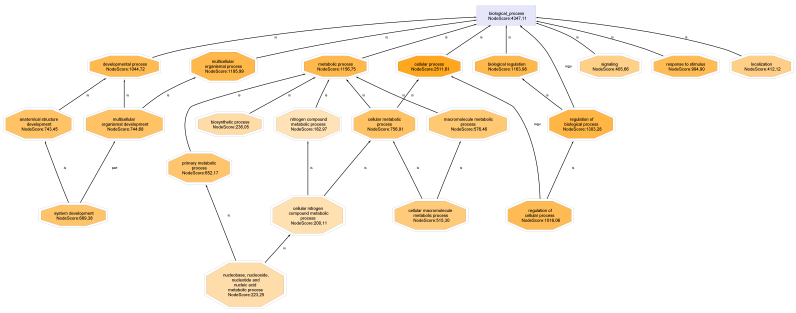


Figure 21: Biological Process Gene Ontology tree of dataset *l500*, using sequence filter 1200.

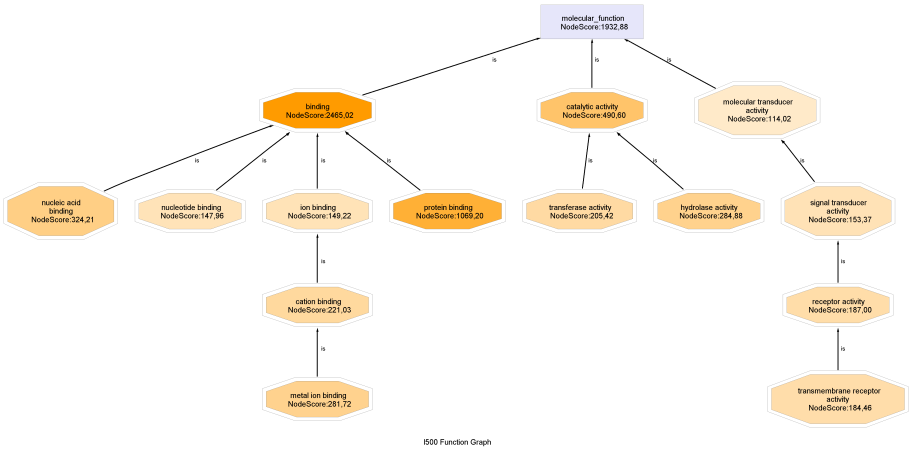


Figure 22: Molecular Function Gene Ontology tree of dataset *l500*, using sequence filter 300.

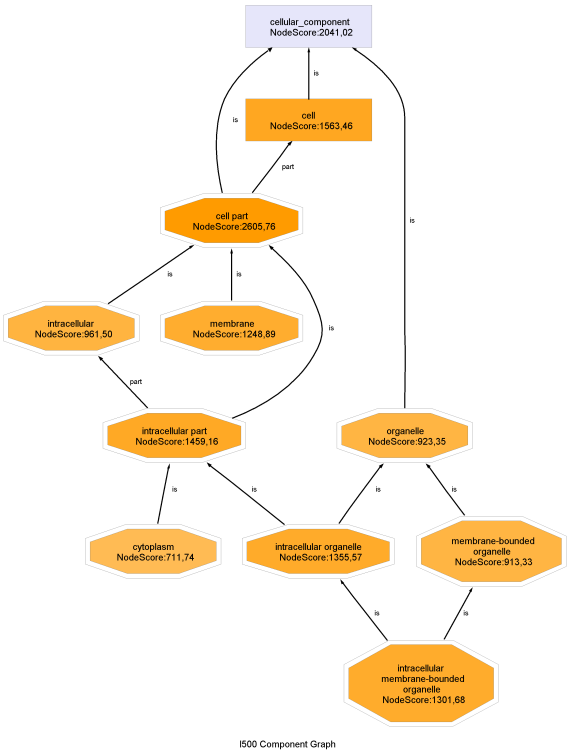


Figure 23: Cellular Component Gene Ontology tree of dataset *l500*, using sequence filter 1300.