LU TP 03-26
May 2003

Bioinformatic Exploration of Antisense Transcripts

Samuel Andersson

Examensarbete för 20 p, Institutionen för datavetenskap och instutitionen för teoretisk fysik,
utfört som en del av masterprogrammet i beräkningsbiologi,
Naturvetenskapliga fakulteten, Lunds universitet

Thesis for a diploma in computer science, 20 credit points, Department of Computer Science and Department of Theoretical Physics, done as a part of the Master's program in computational biology, Faculty of Science, Lund University

Bioinformatic exploration of antisense transcripts
Abstract

Scientists are using microarrays to experimentally measure the expression levels of many individual genes simultaneously. cDNA microarray assays use DNA base-pairing to indirectly measure sequence specific mRNA concentrations, which are believed to reflect gene expression levels. There have been indications that antisense mRNA are more common than expected and it is interesting, and important, to understand how antisense mRNA affects the result of measurements. The aim of this project was two folded; i) to build a software package which create a dynamic database of array clone information (ACID), consisting of complete information of clones used in microarray experiments, and ii) writing an application to find which clones are part of antisense mRNA. Identifying these clones is a first step for further studies of the effect from antisense mRNA.

Most of the information in ACID is collected from well known databases such as UniGene and GenBank. 2368 identified antisense mRNA sequences corresponding to 3032 clones have been found so far. The list can not be seen as final before it is known which clones really are located in the overlapping sense-antisense regions of the mRNA sequences.

Bioinformatisk undersökning av antisense transkriptioner
Sammanfattning

Forskare använder microrarrayer för att på experimentell väg mäta uttrycksnivån för många individuella gener samtidigt. cDNA-microarray använder DNA-basparning för att indirekt mäta koncentrationen av specifika mRNA-sekvenser, vilket antas visa uttrycksnivån för genen. Tidigare studier har indikerat att antisense-mRNA är mer vanligt förekommande än förväntat. Det är både intressant och viktigt att förstå hur antisense-mRNA påverkar resultatet av mätningarna. Målet för detta projekt var tvådelat; i) att bygga ett mjukvarupaket som skapar en dynamisk databas med information om kloner (ACID), innehållandes komplett information om klonerna som används i microarray-experiment, och ii) att skapa en applikation vilken ska finna de kloner som är en del av antisense-mRNA. Identifieringen av dessa kloner är första steget för ytterligare studier om effekterna av antisense-mRNA.

Den största delen av information i ACID är hämtad från välkända databaser så som UniGene och GenBank. 2368 identifierade antisense-mRNA-sekvenser som motsvarar 3032 kloner har upptäckts hittills. Listan kan inte ses som fullständigt innan man vet vilka kloner som verkligen ligger inom det överlappande sense-antisense-området på mRNA-sekvensen.

Contents

Preface

I took this oppurtunity to acknownledge the other people behind this project. I would like to give special thanks to my thesis advisors Jari Häkkinen and Markus Ringnér for helping me in doing this project. I must also thank Srinivas Veerla, my co-partner in developing ACID. Last but not least, I would like to thank Naomi Glarner for sharing her vast amount of knowledge into the world of biology.

Chapter 1

Background

This chapter will give some general information to help understand this thesis, and it starts by explaining the basics of genetics, moving on to describing microarrays, a technique that is used to analyse gene expression. Finally the databases used for information searching are described.

1.1 Basic Genetics

Most of the information of a cell's activity is encoded in the deoxyribonucleic acid (DNA). The DNA is a double stranded linear molecule where the strands are intertwined around each other, like a double helix (Watson and Crick, 1953) (see figure 1.1).

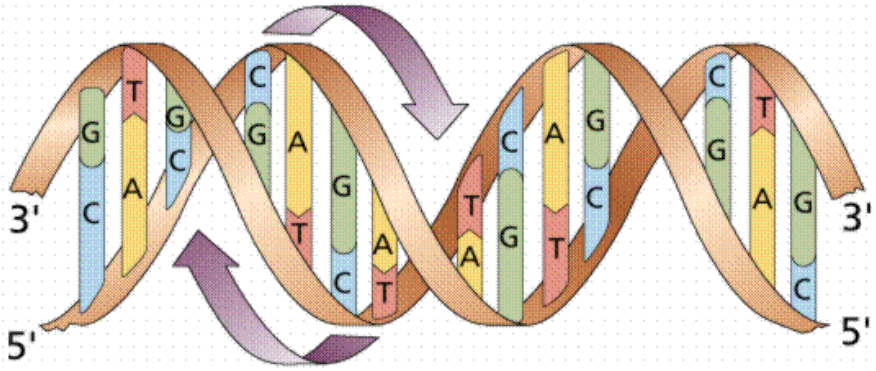

Figure 1.1: *Double-stranded DNA*

DNA is composed of four types of nucleotides bases, adenine (A), thymine (T), cytosine (C) and guanine (G). The strands are kept together by chemical bindings between the bases, where the A binds to T and C binds to G. In this way the two strands are exactly complementary to each other. The DNA can be composed by millions of nucleotides and in cells it is divided into chromosomes. Nucleotide molecules are

assymetric, and each nucleotide of DNA binds to the adjacent nucelotide in a way which makes DNA have a direction with ends (conventionally) called 5' and 3' (see figure 1.1).

A gene is a piece of the DNA sequence which encodes a protein. The DNA strand on which the gene is located is called sense strand, the opposite is called antisense strand. The translation of the genetic sequence into a protein is a two-step process. In the first step the DNA is copied into messenger ribonucleic acid (mRNA) in a process called transcription. The mRNA is a single-stranded copy of the original gene. In the second step the mRNA is translated into an amino acid chain; a protein, during translation. During the translation, three bases compose one codon typically representing one amino acid. Since there are 4 possible nucleotides for each position of the three-letter codon, there are total 4^3 (64) different codons available, which makes the DNA an efficient store of information.

The step from DNA to mRNA is well understood, for every nucleotide in the DNA there will be one added to the mRNA. Therefore the mRNA sequence is an exact copy of the original DNA with the only exception of one base; the thymine (T) will be substituded by uracil (U) in the mRNA. As with DNA, mRNA also has 5' and 3' ends and is read in the same direction as the DNA. Before the mRNA is translated into a protein, certain non-coding regions of the mRNA called introns will be cut out such that only the coding regions called exons are left (see figure 1.2).

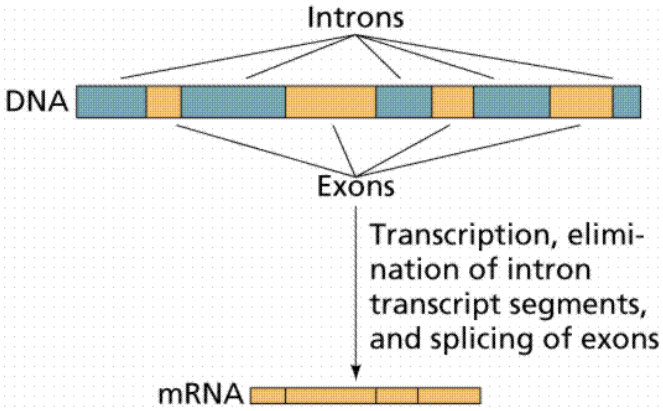

Figure 1.2: *Transcription with splicing*

1.2 cDNA Microarray

Figure 1.3 schematically illustrates the cDNA microarray methodology.

With the cDNA (complementary DNA) microarray technology thousands of gene expressions can be measured at the same time. A gene's expression level is ideally a measure of how much proteins that a gene produces at a given time. What a microarray really measures is the amount of mRNA in the sample, which is extracted from the

tissue of interest. mRNA is an unstable molecule with a short lifespan and can therefore not be used in the microarray process, instead the stable cDNA is used. cDNA is a DNA sequence that is a copied from a RNA template in a process called reverse transcription. cDNA will thus in principle represent a gene.

The experiment process is started with preparing the measuring device, a glass slide, by placing thousands spots of clones (copy of a short portion of DNA), were each spot ideally represent one gene. mRNA is extracted from the tissue under study and the mRNA is reversely transcribed into cDNA and labeled with fluorochromes which makes it emit a specific wavelength. The labeled cDNA is then flushed over the glas slide and all DNA from one gene will bind (hybridize) to one spot. Most of the time there will also be a reference sample flushed over the same slide with a different colour. The reference sample is used to compare two, or more, sample states. Then, by measuring intensity of these colours, the relative gene expression will be known. If a spot shows difference in intensity between the colours, the gene from the sample is not expressed as it is in the reference sample. In that way you can see the difference in gene expression in two samples, e.g. between a healthy cell and a cancer cell.

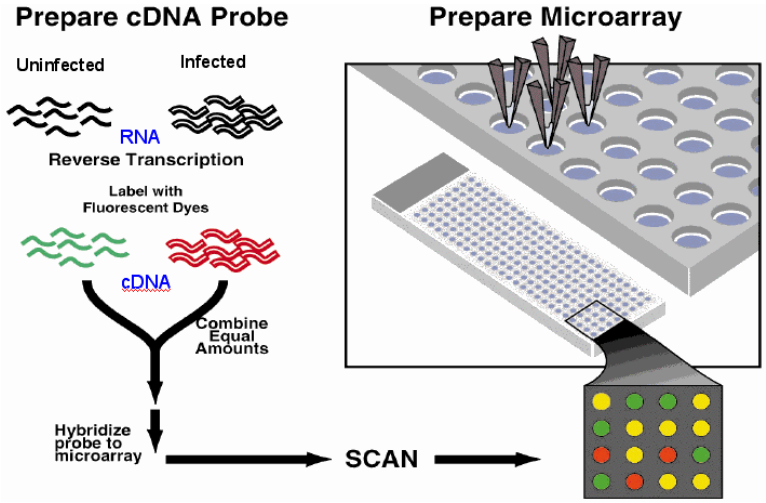

Figure 1.3: *Microarray*

1.3 Databases

Scientists around the world are sequencing and analysing the genomes, complete DNA sequences, of different species to learn more about the species and for research in diseases. The collected information is stored in databases that are shared within the scientific community. In that way researchers have lots of more information to use than if they only searched in the information they collected themselves. The databases are often specialized in some direction, it could be gene annotation, sequences etc.

Chapter 2

Aim of Study

This project had two parts, i) build a database of clone information, and ii) find clones that are part of an antisense mRNA.

2.1 Building Database

When scientists use microarrays they often end up with a large number of genes which are of interest for their experiments. However, to obtain information about the genes one by one may be exhausting. To aid the community in this matter it is of importance to build databases that collects information from well known databases, and present it in such a way that the scientists can search information of clones used in microarray experiments. It is also important to make the application easy to maintain and use.

2.2 Clone Searching

The cDNA clones used in microarray experiments may consist of both the sense and antisense sequence of the gene. That has not been seen as a problem since the common belief was that there were not that many sense-antisense mRNA pairs. Lately there has been experiments that have shown that these pairs are more common than expected[1]. At the Department of Oncology at Lund University there is work going on to construct microarrays where a spot only consist of either the sense or the antisense sequence. The Oncology Department, together with others, have an interest in which of their clones that may be a part of such a pair. By doing a sense-antisense pair search with sequenced mRNAs and then a match against clones, a list of sense-antisense clones can be created.

The mRNAs used in the search can contain both vector sequences and repetitive elements. The vector sequence is a left-over from the cloning procedure. To amplify the clones, the DNA sequence is put into a vector. A vector is a small and circular piece of DNA. These vectors are then put into bacteria and by utilizing these bacteria and vectors the DNA gets replicated. The vectors are isolated from the bacteria and

the inserted sequence is amplified by Polymerase Chain Reactions (PCR). Sometimes the amplified sequence contains a part of the vector's DNA. Repetitive elements are regions of the genome sequence that repeats itself. Such regions can sometimes disturb analysis of the sequence and such regions are usually marked with a special letter or removed.

The aim was to compile lists of clones that corresponds to the sense-antisense mRNA pairs that were identified.

Chapter 3

Methods

This chapter will describe how the database and software of ACID[2] works. The antisense search will be described in the end of this chapter.

3.1 Materials

To collect information of all clones present i the UniGene[3] database, we built Array Clone Information Database (ACID) which contain information collected from publicly available databases.

In UniGene ESTs (Expressed Sequence Tag) are assembled into clusters, where every cluster corresponds to a specific gene. Each cluster has an amount of cDNA clones that is related to the same gene. The sequences of the clones are only partly known. A fraction of mRNA sequence is called EST and has been read in either 3' or 5' direction (see figure 3.1). This information is found in a UniGene cluster. Clusters also contain other information such as the mRNA RefSeq[1][4], the chromosome position, and the gene name. UniGene stores information of many different species.

Sequences for ESTs and RefSeqs are stored in the GenBank[5] database. UniGene clusters only refer to the sequences by GenBank accession numbers. Additional information for RefSeqs, such as chromosome positions, are found in the Genome Browser[6].

Information obtained and stored in ACID for the clones includes:

- GenBank accession numbers to EST sequences
- 3' or 5' direction of EST reads
- which UniGene cluster the clone is related to

[1]The RefSeq (Reference Sequence) collection is aimed to be comprehensive, integrated, non-redundant set of sequences, including genomic DNA, transcript (RNA), and protein products, for major research organisms.

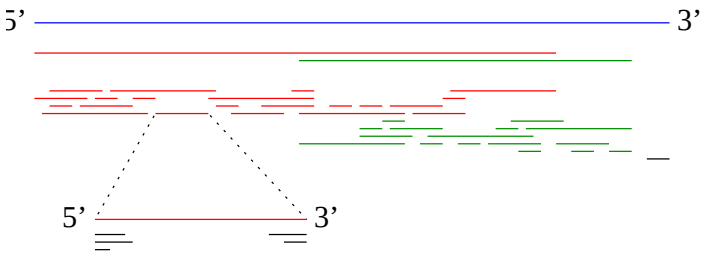

Figure 3.1: *UniGene cluster (long top line, blue) has several ESTs (short lines, red and green) associated to the cluster. These ESTs are also associated to RefSeqs (medium long lines, red and green, below top line, blue) that also are associated to the cluster. The blowup illustrates the partial sequencing of ESTs (and thus clones) where the black lines illustrates 3’ and 5’ reads.*

- chromosome position
- cytoband information
- LocusLink information[5]
- OMIM information[7]
- gene symbol and name
- associated Gene Ontology[8] terms
- associated RefSeq

And information for RefSeqs includes:

- which chromosome it is located on
- start and end base in the chromosome
- cytoband information
- GenBank accession number to sequence
- which UniGene cluster the RefSeq refers to

ACID will be limited to the species present in UniGene and only obtaining RefSeqs that has GenBank accession number that starts with NM, which are the mRNAs.

Homo sapiens (UniGene build 160) and *Mus musculus* (UniGene build 120) were the species put into ACID during my project time. For humans there were 2.5 million clones, 3.2 million sequences and 111 000 UniGene clusters. Of the clusters, there were 18 262 that had at least one related RefSeq. For mouse the numbers were 2.5

million clones, 3.1 million sequences, and 90 000 UniGene clusters of which 12 860 had at least one related RefSeq.

Both UniGene and GenBank information was downloaded from NCBI (National Center for Biotechnology Information) ftp severs

ftp://ftp.ncbi.nih.gov/repository/UniGene.

Gene Ontology information from

ftp://ftp.ncbi.nih.gov/refseq/LocusLink/

The Genome Browser information was retrieved from databases that were stored locally at the department.

3.2 Database Architecture

To keep track of which species ACID holds, a species independent table (**Species**) was created that contains the short name of the species, UniGene build number and the full species name. The species dependent tables' names are prefixed with the short name of the species. Species specific tables will be created automatically when new species are added.

The tables in ACID are the following:

- **Species** table keeps track of the species in the database as described above.
- In **Clones** information about its ESTs are stored. Such as GenBank accession number and whether it is a 3' or 5' read.
- Most of the cluster information is stored in **UniGene** table such as chromosome, gene and cytoband.
- **RefSeqs** contains what cluster a RefSeq belongs to, chromosome start and end, strand etc.
- **GenBank** has all the sequences.
- **Location** is the table that has the clone position within the associated RefSeq. This information is automatically calculated when updating the database. The calculation is done by aligning the clone against the RefSeq using the BLAST[9] program. Since the clones only are stored as some EST sequences and they can have more than one EST for 3' read and 5' read, every combination between 3' and 5' are stored in pairs to make the retrieving for viewing easier and faster. There is also extra BLAST information such as score and expectation value.

The database was built efficient and dynamic to be well adapted to the large amount of data that usually is involved in genetic projects. Each species information in the database is stored in different sets of tables, in order to gain some speed when retriving

or updating the information of the database. Before updating the species information the tables of the specific species are truncated and new information which is stored in files are bulk-inserted. This update process was estimated to take 14 hours for the two species, human and mouse, together. Compared to having the species in the same table and doing deletes, the gain was estimated to 5 hours.

Another problem besides time was how to divide and connect the information between the tables. We could not find any complete description of UniGene. So while building the first version of ACID we got some problems, one was because it turned out that some data had many-to-many relations. And some data did not look as we expected at the beginning, so we had to do a more thoroughly analysis which took lots of our time. To go around problems we made the database somewhat redundant, and furthermore removed multiple RefSeq associations to clusters so that there was at most one for each cluster. The longest RefSeq was the one kept.

The database server used is mySQL (v3.23). The strength is that it is easy to setup and work with. The weakness is that it lacks some SQL features that are common in most other servers.

3.3 ACID Software

The largest part of the software is for updating the information in the database, the other part is the search and presentation with a web interface. My part was to develop the database structure and updating part of the software together with another student. The design decision was to make as much work as possible during updating instead of doing the work while selecting data. The application should also be dynamic to which species that are stored and know which species that need to be updated.

When updating the first thing that is done is checking the build numbers of all the species that are stored in ACID against the the ones in UniGene. For the species where the build number is higher in UniGene, the application downloads the new information and all the sequences that is connected to the clusters. After downloading the information is parsed into files, named after the tables where the information will end up. The files are saved in a proper format to facilitate bulk-insert into the database. The tables that are to be updated are truncated just before insertion.

The **Location** table is the last to be updated, because it needs the data from the other tables to be able to BLAST clones vs RefSeqs. The procedure is to take one RefSeq at the time. Then all ESTs belonging to that RefSeq are BLAST against the RefSeq to get the EST positions within it. BLAST results of ESTs belonging to the same clone are grouped together in such a way that 3' reads are matched to 5' reads in all combinations. Those combinations are then stored in **Location**, together with some BLAST results and idenfication of what was used in the BLAST. For more information about BLAST and how it was used in this project see Appendix A.

Another party developed a web interface from where a user easily can search by

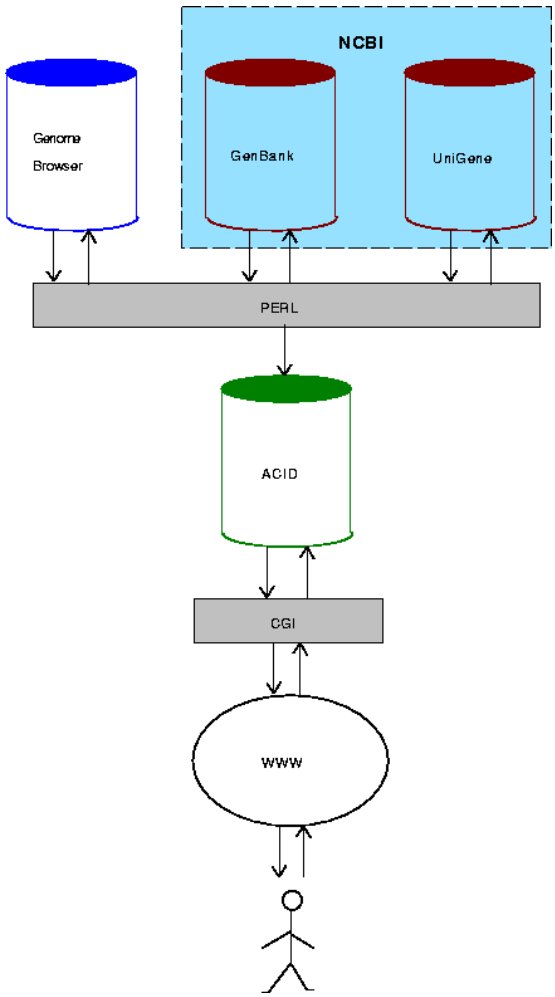

Figure 3.2: *ACID architecture*

clones, RefSeqs, gene names etc. From the search you can choose which information to be retrieved, i.e. Gene Ontology, clone positions and cluster information. See 3.2 for an overview of the ACID architecture.

All software was written in object oriented Perl (v5.8). We made that choice since we where going to perform a lot of textparsing, which Perl is good at, and since Perl is rather common in the bioinformatics community. At the beginning we also used BioPerl, a Perl package for bioinformatics. But after a long struggle with memory leakage in BioPerl we threw that package out of our project and wrote our own code. We also noticed that Perl had some minor memory leakage when using the built in hash, so we minimized the usage of hash variables in our project.

3.4 Prototyping a Sense-Antisense Search Application

To save time in this second part, I decided to develop small programs to solve the different steps in the search for antisense RefSeqs and find clones included in those RefSeqs. In that way the development was faster and I could do more within the project timespan. I used some BLAST applications and did some small Perl scripts. These programs can be used to develop a full application in the future.

The first step was to create a FastA formated file with all the RefSeq sequences that was to be tested. A FastA formated file is a file of sequence entries where every entry starts with a header that has an identification for the sequence and can have some additional information, the line with the header starts with '>'. The sequences comes on a new line directly after the header. The headers I created only contained an identification, a GenBank accession number for the sequence, like this

```
> NM_002223
ATCCCAGGTTACTTACCAATTAA
TTCCAAAGGATAAAAACCCGGTT
...
```

I wrote a small Perl script that fetched all the RefSeqs that had sequences in ACID and stored those in a FastA formated file. The next step was to filter the RefSeq sequences from vector sequences. A FastA formated file with vector sequences was downloaded from *ftp://ftp.ncbi.nih.gov/pub/UniVec/*. To be able to use the BLAST tools needed for this project, a BLAST prepared database had to be created, and that is done with formatdb. A database was built of the vector sequence file; the purpose was to use it with the BLAST tool vecscreen. vecscreen is an application that takes a FastA formated file and BLASTs it against a database, in this case the vector database. The result of the BLAST was then analysed to see if there is any significant vector hits in the sequence. For the project there were three different situations that were of interest. 1) If vecscreen spits out the result "No significant similarity found", then the sequence was kept. 2) If the result says that it had matching segments and one of the segments was in the middle of the sequence, the sequence was rejected. 3) If the matching segments only appear in the end of the sequence, the segments were cut off and the rest of the sequence was kept. A Perl script was created to do the vecscreen run for each RefSeq and doing the cutting if needed. All the RefSeq sequences that went through that filtering was saved into a new FastA file. The last step I had time to do before looking for matching clones was to do the BLAST to find antisense hits within the RefSeq set. For this, BLAST requires a database to search against. Since all RefSeqs are compared to their reverse complementary sequences, I created a BLAST database with the RefSeqs without vector sequences. Then all complementary RefSeqs were BLAST against this database. The tool blastall was used for that and the result file was then parsed by yet another Perl script. The Perl script takes all the RefSeqs that have hits and the best hit of those hits, including BLAST score and e-value for that hit,

and store it in a file. The RefSeqs in that file were said to be antisense RNA.

We have a list of all clones available for us through the Department of Oncology. From that file all clone identifications were parsed out, and a search was made in ACID to find which RefSeqs matches the clones. By matching those RefSeqs to the ones that was found to be antisense, a list of clone-RefSeq pairs was stored in a file to be used for further investigation in the future.

For more information of how the different BLAST applications were used, see Appendix A.

Chapter 4

Results

ACID is up and running, and is already used by the scientific community in Lund. The aim for ACID was to make it easy to maintain and use. The update procedure is in principle only one function call, so it is easy to make a script to run in the background. The system will only really update if there are new builds for the species included in the database. The system is currently configured for *Homo sapiens* and *Mus musculus*. With some minor changes it will support more species, because only a few lines of code have to be added to support more species. The application is also easy accessible by the aid of the web interface. This interface allows the user to send in a bulk of clones and get the result of all of them either by viewing them directly in the browser or save it to a nicely formated file.

The tools I developed for sense-antisense search are prototypes and can be seen as initial building blocks of an application. These initial building blocks include the vector filtering and the search part. Using that tools I found 2368 RefSeqs that had an antisense partner and 3032 clones belonging to those RefSeqs. My results agree with the article *Antisense transcripts in the human genome*[10]. The database in that article had 12 897 RefSeqs compared to ACID's 15 666, the article had a result of about 1 800 RefSeqs with an antisense partner.

Chapter 5

Conclusion and Future Prospects

ACID is a useful system and hopefully it will grow in amount of species included into the database, and in number of users. An installation script would probably be nice to have even if it is relatively easy to install as it is. The species in ACID are hard coded to human and mouse. That hard coded part of ACID should be changed to be more dynamic, so that there would not be any need to change in the code when adding new species.

It was sometimes frustrating to develop this size of project in Perl, not just because of the bugs in Perl and BioPerl, but also the lack of stronger typing and better support of object oriented programming. A more powerful language in that sense would have made the coding a bit smoother. When coding smaller test programs and parsing parts Perl felt a lot more flexible and powerful. With all the free packages out there it was easier to concentrate on the project's problems instead of some technical issues such as implementing protocols.

2368 RefSeqs of 15 666 seems like a lot, if the sense and antisense genes are expressed at the same time it can have significant impact on the result of the experiment. By having a list of clones which are part of the sense-antisense mRNA pairs, the sense-antisense experiments can be narrowed down to test only clones from the list. To be really sure which clones to use, a search to find clones that are in the overlapping part of the sense-antisense pair are of interest. The current application compiles a list of clones corresponding to mRNAs that may have repeats. As soon as the tool[11] to identify, and remove, repeats sequences from mRNA becomes available for local download, the application should be used to compile a corresponding list of clones for these non-repeat mRNAs.

Bibliography

[1] *Do natural antisense transcripts make sense in eukaryotes?*, Vanhée-Brossollet C and Vaquero C (1998). Gene 211:1-9

[2] *ACID: a database for microarray clone information*, Ringnér M, Andersson S, Veerla S, Staaf J and Häkkinen J (2003). LU TP 03-24, Lund University.

[3] *NCBI UniGene* [http://www.ncbi.nlm.nih.gov/UniGene]

[4] *RefSeq and LocusLink: NCBI gene-centered resources*. Pruitt KD and Maglott DR (2001). Nucleic Acids Research 29:137-40

[5] *GenBank*. Benson DA, Karsch-Mizrachi I, Lipman DF, Ostell J, Wheeler DL (2003). Nucleic Acids Research 1:31:23-7

[6] *The Human Genome Browser at UCSC*. Kent WJ, Charles W Sugnet, Terrence S Furey, Krishna M Roskin, Tom H Pringle, Alan M Zahler and David Haussler (2002). Genome Research 12:996-1006.

[7] *Online Mendelian Inheritance in Man OMIM: www.ncbi.nlm.nih.gov/entrez*. Parton MJ (2003). Journal of Neurology Neurosurgery and Psychiatry, 74:703

[8] *Creating the gene ontology resource: design and implementation*. The Gene Ontology Consortium (2001). Genome Research, 11:1425-1433.

[9] *NCBI BLAST* [http://www.ncbi.nlm.nih.gov/BLAST]

[10] *Antisense transcripts in the human genome*, Lehner B, Williams G, Campbell RD and Sanderson CM (2002). Trends in Genetics 18:63-5

[11] *RepeatMasker* [http://ftp.genome.washington.edu/cgi-bin/RepeatMasker]

Appendix A

BLAST

In this project BLAST (Basic Local Alignment Search Tool) was used for doing sequence alignment. BLAST is more or less a standard tool for the biology community when working with sequences, whether it is protein or DNA. The BLAST applications used in this project were formatdb, bl2seq, blastall and vecscreen. formatdb, bl2seq and blastall can be downloaded from
ftp://ftp.ncbi.nih.gov/blast/executables
vecscreen has to be compiled from the source that can be found at
ftp://ftp.ncbi.nih.gov/toolbox/ncbi_tools

bl2seq
ACID uses bl2seq when performing comparison between RefSeq and EST sequences. The result from bl2seq was parsed and stored in the **Location** table. When working with this application all sequences and the results were stored in temporary files. The parameters I used were

```
bl2seq -i refseq -j estseq -p blastn -o result
```

Were *refseq* and *estseq* is sequences to compare, *result* is the result and *blastn* tells bl2seq that it is DNA sequences to compare.

formatdb
Most BLAST applications are performing comparisons against databases with sequences. To build such BLAST database formatdb is used. In the search for antisense mRNA, database searches were necessary and were created with these parameters

```
formatdb -i fastafile -p F -o T -n dbname
```

fastafile is a FastA formated file with sequences, *F* flag tells that the file contains DNA and the flag *T* is set to make formatdb create indexes for the sequence identifications. The last parameter is the base name for the database files is set to *dbname*.

blastall
blastall takes a FastA formated file and compare all entries in it against a BLAST database. This application was used to search for antisense mRNA. The file and the database contained the same sequences and only searched for complementary sequences.

```
blastall -p blastn -d dbname -i refseqs -o result -e 1e-9 -S 2 -b 0 -v 3000
```

blastn tells the program to perform DNA comparison, *dbname* is the database to work against, *refseqs* is FastA formated file to compare with and *result* is name of the file to save BLAST result in. Parameter *-e 1e-9* sets the Expect cut off to 10^{-9}, that is limit to how bad an alignment can be before rejected as a hit. *-S 2* will make blastall to only do complementary comparison. To make the result file smaller no alignments were saved by setting show numbers to 0. Last thing to set is the number of hits to show, it is set 3000 to be sure that the best hit always is in the result list.

vecscreen
In search for vector contamination in sequences vecscreen is used. The result marks regions suspected to have vector sequence. The sequence to be tested was piped to vecscreen therefore the lack of infile.

```
vecscreen -o result -d dbname -f 1
```

result is were there result is saved and *dbname* is the database to search in. By setting -f is the output format and 1 is for HTML format with no alignments.