Final Thesis

On text mining to identify gene networks with a special reference to cardiovascular disease.

Per Erik Strandberg

LiTH - IFM - EX - 05 / 1382 - SE

On text mining to identify gene networks with a special reference to cardiovascular disease.

IFM - The Department of Physics and Measurement Technology, Biology and Chemistry, Linköpings Universitet

Per Erik Strandberg

LiTH - IFM - EX - 05 / 1382 - SE

Examensarbete: **20 p**

Level: **D**

Opponent: **Andreas Lundgren**

Supervisor: **Jesper Tegnér**,
IFM - The Department of Physics and Measurement Technology, Biology and Chemistry, Linköpings Universitet.

Supervisor: **Johan Björkegren**,
CGB - The Center for Genomics and Bioinformatics, Karolinska Institute.

Examiner: **Bengt Persson**,
IFM - The Department of Physics and Measurement Technology, Biology and Chemistry, Linköpings Universitet

Linköping: **February 2005**

LINKÖPINGS UNIVERSITET

Avdelning, Institution
Division, Department

IFM - Linköpings Universitet
SE-581 83 Linköping
SWEDEN

Datum
Date

February 2005

Språk
Language

☐ Svenska/Swedish
☒ Engelska/English
☐ _____

Rapporttyp
Report category

☐ Licentiatavhandling
☒ Examensarbete
☐ C-uppsats
☐ D-uppsats
☐ Övrig rapport
☐ _____

ISBN

ISRN
LiTH - IFM - EX - 05 / 1382 - SE

Serietitel och serienummer
Title of series, numbering

ISSN

URL för elektronisk version
http://www.ifm.liu.se/∼perst/

Titel
Title

On text mining to identify gene networks with a special reference to cardiovascular disease.

Författare
Author

Per Erik Strandberg

Sammanfattning
Abstract

The rate at which articles gets published grows exponentially and the possibility to access texts in machine-readable formats is also increasing. The need of an automated system to gather relevant information from text, text mining, is thus growing.
The goal of this thesis is to find a biologically relevant gene network for atherosclerosis, the main cause of cardiovascular disease, by inspecting gene cooccurrences in abstracts from PubMed. In addition to this gene nets for yeast was generated to evaluate the validity of using text mining as a method.
The nets found were validated in many ways, they were for example found to have the well known power law link distribution. They were also compared to other gene nets generated by other, often microbiological, methods from different sources. In addition to classic measurements of similarity like overlap, precision, recall and f-score a new way to measure similarity between nets are proposed and used. The method uses an urn approximation and measures the distance from comparing two unrelated nets in standard deviations. The validity of this approximation is supported both analytically and with simulations for both Erdös-Rényi nets and nets having a power law link distribution. The new method explains that very poor overlap, precision, recall and f-score can still be very far from random and also how much overlap one could expect at random. The cutoff was also investigated.
Results are typically in the order of only 1% overlap but with the remarkable distance of 100 standard deviations from what one could have expected at random. Of particular interest is that one can only expect an overlap of 2 edges with a variance of 2 when comparing two trees with the same set of nodes. The use of a cutoff at one for cooccurrence graphs are discussed and motivated by for example the observation that this eliminates about 60-70% of the false positives but only 20-30% of the overlapping edges. This thesis shows that text mining of PubMed can be used to generate a biologically relevant gene subnet of the human gene net. A reasonable extension of this work is to combine the nets with gene expression data to find a more reliable gene net.

Nyckelord
Keyword

Atherosclerosis, Cardiovascular Disease, Cooccurrence, Data mining, Gene networks, Literature networks, Prior incorporation, Text mining.

Abstract

The rate at which articles gets published grows exponentially and the possibility to access texts in machine-readable formats is also increasing. The need of an automated system to gather relevant information from text, text mining, is thus growing.

The goal of this thesis is to find a biologically relevant gene network for atherosclerosis, the main cause of cardiovascular disease, by inspecting gene cooccurrences in abstracts from PubMed. In addition to this gene nets for yeast was generated to evaluate the validity of using text mining as a method.

The nets found were validated in many ways, they were for example found to have the well known power law link distribution. They were also compared to other gene nets generated by other, often microbiological, methods from different sources. In addition to classic measurements of similarity like overlap, precision, recall and f-score a new way to measure similarity between nets are proposed and used. The method uses an urn approximation and measures the distance from comparing two unrelated nets in standard deviations. The validity of this approximation is supported both analytically and with simulations for both Erdös-Rényi nets and nets having a power law link distribution. The new method explains that very poor overlap, precision, recall and f-score can still be very far from random and also how much overlap one could expect at random. The cutoff was also investigated.

Results are typically in the order of only 1% overlap but with the remarkable distance of 100 standard deviations from what one could have expected at random. Of particular interest is that one can only expect an overlap of 2 edges with a variance of 2 when comparing two trees with the same set of nodes. The use of a cutoff at one for cooccurrence graphs are discussed and motivated by for example the observation that this eliminates about 60-70% of the false positives but only 20-30% of the overlapping edges. This thesis shows that text mining of PubMed can be used to generate a biologically relevant gene subnet of the human gene net. A reasonable extension of this work is to combine the nets with gene expression data to find a more reliable gene net.

Keywords: Atherosclerosis, Cardiovascular Disease, Cooccurrence, Data mining, Gene networks, Literature networks, Prior incorporation, Text mining.

Acknowledgements

We would like to thank all who contributed to this study: all the people of the computational biology, computational medicine and bioinformatics groups at IFM and KI - in particular Anders Bresell, Guiyuan Lei and of course José M Peña; my opponent Andreas Lundgren who helped me with excellent critical but constructive feedback and linguistic comments; my supervisors Jesper Tegnér and Johan Björkegren and examiner Bengt Persson for pushing me in the right directions; and of course my cooccurring cohabitee Anna Hultén deserves many thanks for TLC and for making me eat, sleep and shower.

We wish to acknowledge the help of the following software, programming languages et cetera: Cytoscape, Emacs, Firefox, Graphviz, MySQL, Python, ...

People, organizations and enterprises behind important data, like abstracts [PubMed], gene nets [Guelzim et al, Lee I et al, Lee TI et al, Luscombe et al] and synonym lists [Euroscarf, HGNC, SGD] used in this thesis are worthy of a big "thank you guys" from us for having this data publicly available on the web.

Contents

So this is a Harvard bar.
I thought there'd be equations and shit on the walls.
Good Will Hunting

1 About this text

THIS TEXT IS WRITTEN AS a master of science final thesis at Linköpings Universitet by Per Erik Strandberg with Jesper Tegnér and Johan Björkegren as supervisors and Bengt Persson as examiner, in the autumn of 2004. This report covers one way to find interesting genes and gene networks for the largest health issue we know: cardiovascular disease or more specifically atherosclerosis. This report is part of a greater study conducted by members of Linköpings Universitet and Karolinska Institutet in collaboration. Some of the overall goals is to find regulatory pathways, responsible genes and with them evaluate and propose new treatments.

Below is a short list of the main chapters in this thesis, and what they treat.

Introduction: Background, motivation and goals: A brief background is given, explaining why gene nets are relevant, how scientists have found them in the past and what we want to achieve in this thesis.

Tutorial: Some concepts occurring throughout the text are explained. Experienced readers might want to skip this chapter, butt readers with little or no experience of text mining should probably read this chapter carefully. Of particular interest is the section explaining the urn and the section discussing P, R and F and the examples in the end of each of them. The extension of the two examples is what gives birth to a new validation method used in later chapters.

Methods: This chapter explains how the abstracts (the indata) are treated and how we can go from abstracts to cooccurrences and from cooccurrences to gene nets. For readers with little background in this area, this chapter (in combination with the tutorial) is especially important in order to understand this thesis.

Validation and results: This chapter studies the gene nets we now have generated and compares them with previous studies. In order to perform this validation a new tool for doing pairwise comparisons of graphs is created. In addition to the final chapter this is the most interesting chapter of this thesis.

Summary, conclusion and discussion: A summary of this thesis, with conclusions and a discussion of how we can interpret the results.

Appendix: An appendix with some algorithms, psuedocode and database-schemas.

Why did you put that weapon together so quickly, Gump?
You told me to, Drill Sergeant.

Forrest Gump

2

Introduction: Background, motivation and goals

CARDIOVASCULAR DISEASE is the cause of almost one in two deaths in the USA. Atherosclerosis (explained in next chapter) is the main cause of cardiovascular disease. Much effort has been put into understanding this condition since describing and explaining how and why atherosclerosis occurs on a gene level is one of the more important tasks for humanity in the postgenomic era. This final thesis is one part of a larger study in which we want to use different methods to understand atherosclerosis. One way of understanding this is to find a relevant gene network of the disease.

When studying a large system it is often a good idea to study a part of it. As a metaphor we could imagine traffic fluctuations. If we understand how a car behaves it is easier to understand how the traffic behaves on a road and in a city. It is now probably easy to model it in many other cities and even in a country or state. In a similar way a gene net can be used to understand life: In order to understand life one might want to study an organism, in order to understand an organism one might want to study one of its cells, and in order to understand a cell one might want to understand a gene or a net of genes. It is therefore very important to understand and explain gene networks in order to understand and explain the dynamics of life. Another equally important reason of understanding gene networks is to understand, explain, predict and remedy disease. If we go back to the metaphor of traffic it is not hard to imagine that if we understand why and how traffic jams and accidents occur we might understand what could be done to avoid them - perhaps by leading traffic via other roads or closing some that are dangerous. By studying gene nets we do not only understand the normal functions of life, potentially we could also understand defect states better, such as cardiovascular disease.

If we had the gene network of an organism or a disease we might ease the work of pharmaceutical companies in their search for drugs. Sometimes such companies have lists of how old drugs affect the state of genes in an organism or a cell, but they do not necessarily have the gene network they are trying to attack. If they were to have this gene network they could perform a more coordinated research and potentially combine drugs to target genes in a specific path and understand what genes that are the primary affected genes and which ones that are affected in a secondary way.

Scientists have for some time used molecular biology to study how genes interact with other genes. Two such methods are (i) microarray studies where the expressed genes are measured to get information about which genes that are active in a cell or tissue and (ii) chip-chip (Chromatin Immunoprecipitation chip) techniques that are used

to investigate interactions between proteins and DNA. Chip-chip is typically used on yeast. A complement to these traditional methods is text mining.

One definition of text mining referred to in [Shatkay and Feldman] is “the combined, automated process of analyzing unstructured natural language text in order to discover information and knowledge that are typically hard to retrieve”. As opposed to microarray techniques text mining is cheap: all one needs is a computer and some data. The data, in this thesis abstracts from PubMed [PubMed], is often free and available from the internet. A second reason to use text mining is that it is quite fast - once a system is up and running it is easy to add abstracts and scan for genes in different subsets in it, there is the possibility to try almost any hypothesis. Text mining is more and more becoming a complement to traditional methods, but will probably never be able to replace molecular biology. One of the early large-scale text mining projects is described in [Jenssen et al].

The goal of this thesis is to find a biologically relevant gene net for atherosclerosis by investigating gene cooccurrences in PubMed. In order to do this we will: (i) Implement an automated system doing text mining. (ii) From this automated system gather gene networks for yeast and show that these nets are relevant compared to other gene nets for yeast - possibly motivating that text mining is a valid method. (iii) Evaluate results from atherosclerosis related scans by comparing hit lists of genes with other hit lists.

The implementation of the automated system is explained in the Methods chapter and the analysis of the nets and hit lists in the Validation and results chapter. There is also a short Summary, conclusion and discussion chapter.

It's talking Merry. The tree is talking.
Tree? I am no tree! I am an Ent.

The Two Towers

3 Tutorial

THIS TUTORIAL WILL MENTION and define some concepts later used in this thesis. Concepts an experienced reader might already be familiar with and want to skip. Some concepts discussed are: atherosclerosis, abstract, cooccurrence, graphs and the statisticians best friend: the urn. Of particular importance are the two sections at the end and the examples in each of them, since this is a nice illustration and motivation for the method developed to perform pairwise comparisons of graphs.

3.1 What is atherosclerosis?

Cardiovascular disease is the cause of almost one in two deaths in the USA and atherosclerosis is the main cause of cardiovascular disease. This is more than the one in three deaths that cancer cause. Atherosclerosis is a complex gradual process that occurs when cholesterol accumulates under the inner lining of artery walls due to damage from uncontrolled high blood pressure, smoking, diabetes, or high blood cholesterol. The deposits (cholesterol plaques) eventually result in fibrosis and calcification, which may narrow or block the artery and hinder blood flow. Atherosclerosis is also called "hardening of the arteries". The disease can produce a chest pain when some part of the heart does not receive enough blood (angina pectoris), heart attack, or stroke. [AHA, Hansson, Humphries and Morgan, Libby, Lusis]

3.2 What is an abstract?

An abstract is a summary of an article written to allow for readers to get a condensed version of a text. PubMed, [PubMed], a service of the National Library of Medicine, includes millions of citations and almost as many abstracts for biomedical articles and allows users to download abstracts from the internet for free. The information is available in many formats, such as XML or a more readable format called "medline" as illustrated in figure 3.1. This figure also shows the kind of information given in each abstract.

3.3 What is a cooccurrence?

The text mining in this thesis has a simple and almost naive idea: if more than one gene name occur in an abstract (or title) we say that each occurring pair in the abstract is a cooccurrence. The cooccurrences are believed to reflect biological functions

```
PMID-14644386
DA  -20031203
DCOM-20040826
DP  -2003 Dec
TI  -Effects of 3-deazaadenosine on homocysteine and atherosclerosis in
     apolipoprotein E-deficient mice.
AB  -OBJECTIVE: In the past decade, elevated homocysteine concentration has
     achieved widespread recognition as an independent risk factor in the
     development of atherosclerosis. 3-Deazaadenosine (c3Ado) is a potent
     inhibitor and substrate for S-adenosylhomocysteine hydrolase and therefore
     may reduce homocysteine concentrations. The current study investigated the
     effect of c3Ado on serum homocysteine, atherosclerotic lesions, and the
     expression of adhesion molecules in apoE-knockout mice. METHODS AND
     RESULTS: Animals were placed on an atherogenic diet with or without c3Ado
     for 12 and 24 weeks. Frozen cross-sections of the aortic sinus and the
     proximal aorta were analyzed by computer-aided planimetry for fatty plaque
     formation. Macrophages, VCAM-1 and ICAM-1 were quantified by
     immunhistochemistry and oligo-cell reverse transcription polymerase chain
     reaction after laser microdissection. Application of c3Ado resulted in
     significant reduction of homocysteine levels by 35.9 and 45.3% after 12
     and 24 weeks, respectively (P < 0.001). Neointimal area and
     atherosclerotic plaque formation were significantly reduced in animals
     treated with c3Ado (P < 0.01). Moreover, monocyte adhesion and concomitant
     ICAM-1 and VCAM-1 antigen and RNA expression on the endothelial layer were
     significantly reduced (P < 0.001, P < 0.01). CONCLUSION: Our results
     demonstrate that c3Ado induces a marked reduction of homocysteine
     concentrations which might explain in part the anti-atherogenic effect of
     the drug.
```

Figure 3.1: Selected parts of an abstract. Notice that there are “fields” like PMID, TI and AB for PubMed identifier, title and abstract. The full abstract contains a lot more information, like authors and so on.

[Jenssen et al]. We let a cooccurrence be displayed graphically by an edge between two nodes in a network, like in figure 4.4. Here the nodes are the cooccurring genes and the edges corresponds to cooccurrences and hopefully some biological function. By doing this for many abstracts we quite rapidly get a large net.

3.4 What is a gene expression matrix?

Gene expression is the process by which a gene’s coded information is converted into the structures present and operating in the cell. Expressed genes include those that are transcribed via RNA to proteins and those that are transcribed into RNA but not translated into proteins.

When measuring gene expression one often measure some kind of concentration of a genes RNA, for many genes in many samples. What you get can be described as a matrix where a column corresponds to a certain gene, and each row is a sample as described in table 3.1. Traditionally the number of genes studied is much larger than the number of samples.

3.5 What is a graph?

A graph can be described by two sets. The first, often called N, is a set of nodes. The second, often called E, is a set of edges where each edge “connects” two nodes and an edge can thus be considered to be a pair of nodes. In this text the nodes are genes or sometimes MeSH terms. Graphs are often referred to as networks, nets, trees (and not

	gene1	gene2	gene3	gene4	...	geneN
sample1	54	49	87	12	...	52
sample2	64	41	97	16	...	98
⋮	⋮	⋮	⋮	⋮	⋱	⋮
sampleM	44	64	77	11	...	58

Table 3.1: An illustration of how a gene expression matrix could look.

ents), et cetera. More about graphs can be found in [Björn and Turesson], for example. There are also many synonyms for edges and nodes and the general idea of what, for example, a tree is is not constant in for example the Bayesian and computer science communities. We here present a short list of some important concepts and terms and explain how they are used in this thesis:

Directed acyclic graph: a directed graph has no undirected edges and is acyclic or a DAG if it is impossible to start in a node and follow edges back to the initial node. In graph (iii) in figure 3.2 there is a cycle: CDAC, it is thus not acyclic. Graph (iv) is directed and acyclic, it is thus a DAG.

Tree: as for graphs in general, trees can be directed or undirected. A graph is a tree if it contains no cycles. In trees there are as many edges as there are nodes minus one. In a spanning tree all nodes are connected. (Often a set of trees is referred to as a forest.) If we removed one of the edges in the cycle ABCDA in (i) in figure 3.2 it would be a tree.

The degree, or sometimes cardinality, of a node is the number of edges it is included in. For directed graphs there is a difference between indegree and outdegree. Often a node with a large degree is called a hub.

Link distribution: a curve or scatterplot showing the number of nodes having a certain degree, or as a mathematician would say: number of nodes as a function of degree.

A complete graph is a graph in which all nodes are connected to all other nodes. Like (ii) in figure 3.2.

Erdös-Rényi random graphs (ER graphs) are a classic class of random graphs. In this class all potential edges have the same probability of being an edge in the graph. This is equivalent to say that all possible pairs of nodes have the same probability of being connected. By varying this probability the characteristics of the graph dramatically change. Typically ER graphs have no leaves and no extremely connected nodes, instead the majority of nodes have about the same degree [Barabási].

Preferential attachment, another class of graphs could be said to have "preferential attachment". This can be interpreted in the following way: when we construct a graph, we add edges one by one and they act as if they want to be attached to nodes with a high degree. Typically these graphs have many leaves and a few extremely connected nodes. The link distribution of these graphs is linear in log-log-scale.

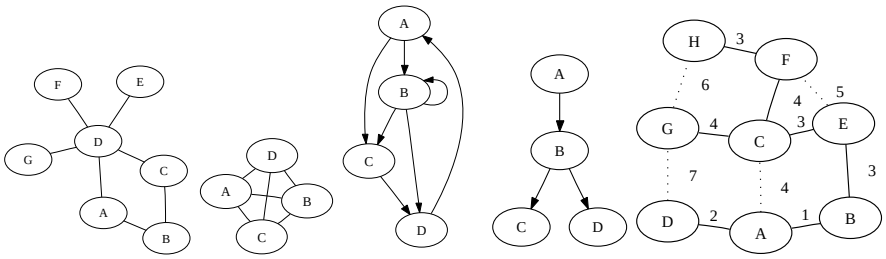

Figure 3.2: Five graphs illustrating some important concepts. (i): the leftmost shows an undirected graph with 7 nodes and 7 edges. (ii): the second graph is a complete graph with four nodes, often called K_4. (iii): a directed cyclic graph. (iv): a directed acyclic graph (DAG), this is also one of many possible trees of (iii). (v): the rightmost graph has weights on its edges. A minimal (cheapest) spanning tree is indicated with full edges.

3.6 What is a MeSH term?

The abstracts one might download using PubMed are often provided with additional information such as MeSH terms. MeSH (Medical Subject Headings) is a controlled vocabulary providing consistent terminology for concepts covered by the database.

In MeSH there are many DAGs, each corresponds to a general concept. The first three DAGs are Anatomy (A), Organisms (B) and Diseases (C). There is a total of 15 such main DAGs. In general the child of a node in these DAGs has a “is-a” or a “part-of” relation to its parents. As an example the first three subDAGs under Anatomy are Body Regions (A01), Musculoskeletal System (A02) and Digestive System (A03).

Each node in MeSH could be considered to be a MeSH term since they are words. Each node may have parents and children. A fix MeSH term may also appear in many DAGs: The MeSH term Mouth, for example, appears in three subDAGs of the DAG A (anatomy): Body Regions (A01), Digestive System (A03) and Stomatognathic System (A14). In Body Regions the node Mouth is a child to Face, grandchild to Head and has the child Lip. In the other subDAGs Mouth has other parents and other children. In addition to this a node may also have additional information not used in this thesis. For more details of and to download MeSH: www.nlm.nih.gov/mesh/ .

In this thesis we have flattened MeSH into one big DAG where a certain term is considered to be one node and thus has all possible parents and also all possible children. By doing this the selection of children and parents of a term was simplified. An example of a subDAG of this big DAG is displayed in figure 3.3.

3.7 What is a gene network?

There is no such thing as direct communication between genes. So a valid question is “why create gene nets if there are none?”. A gene net is a simplification and to motivate it a bit of background is needed: Life as we know it is built up by multi- oligo- or monocellular organisms. In cells information is stored in DNA. Some DNA, genes, are expressed and transformed into RNA and later much of the RNA is translated into proteins. We also know that much RNA is part of proteins such as ribosomes and that

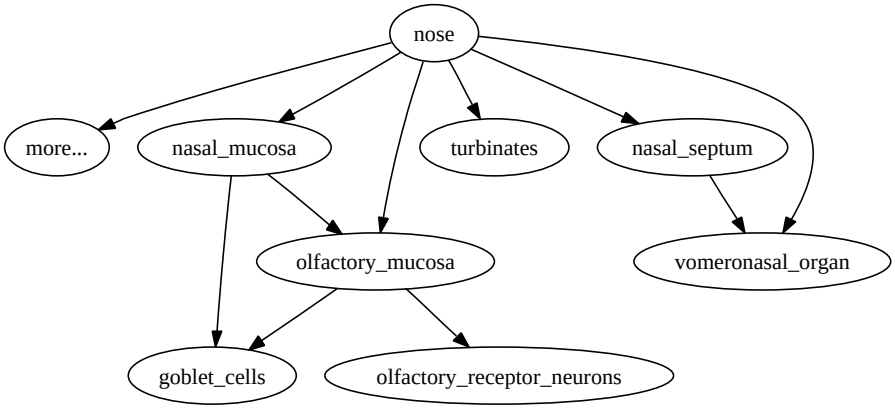

Figure 3.3: The subDAG of nose from MeSH. Please notice that some nodes have multiple parents and that a child often has a “is-a” or “part-of” relation to its parent(s). The MeSH term nose also has more children than displayed, indicated with a “more…” node. The parents of nose are not displayed.

proteins can interact with proteins, RNA and also with genes. A protein that affects the expression of a gene is often referred to as a transcription factor (TF). There is also evidence of RNA interactions with genes. But: genes do not interact directly with other genes. The approximation - that genes interact with other genes - is however not nonsense: there is often an injective relation between a gene and its gene product. If for example a gene codes for a TF, this TF will bind to another gene and regulate its activity. In this case we make an abstraction: we eliminate the TF in between the genes and say that the genes have a biologically relevant link between them. Sometimes this link is interpreted as a directed edge in a gene net, but in this thesis the direction is ignored. This way of interpreting how a gene network could be relevant is explained in figure 3.4.

There are many different types of nets including genes. Below is one interpretation and classification of them:

Gene nets, is a general class of gene nets where any set of genes may have connections.

Gene-protein nets, is a class of nets where a directed edge between a gene and a protein could mean that the gene is expressed as the protein and an arrow from a protein to a gene could mean that the protein regulates the expression of the gene.

A TF net can be a net with only genes and where an arrow from a gene to another means that the first gene encodes a TF that regulates the activity of the second gene. In these kinds of nets a loop would mean that the gene regulates itself.

Cooccurrence nets include the nets generated in this thesis. An edge between two genes mean that they have both occurred in, for example, the same abstract. The weight of the edge is interpreted as the number of times the pair has cooccurred. This kind of net is believed to be biologically relevant.

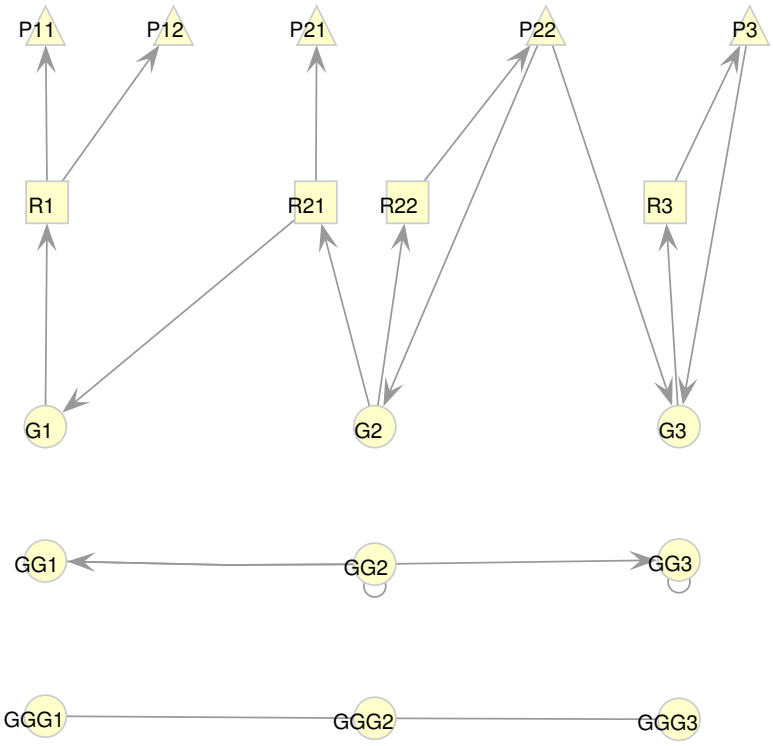

Figure 3.4: In cells genes (circles in the figure) are expressed into RNA (squares in the figure) - illustrated by a directed arrow from the gene to the RNA. Some RNA is translated into proteins (triangles in the figure) - illustrated by a directed arrow from RNA to protein. Other RNA is however functional in different ways and may affect genes, proteins or other RNA - again illustrated with a directed arrow. The uppermost figure is an illustration of this. The middle figure here could be considered a projection of the upper figure into the gene domain, here GGn corresponds to Gn for n= 1,2,3. We can see that every path from a gene to another gene results in a directed arrow in this domain. Notice that there are loops for GG2 and GG3. The lower figure is another projection of the uppermost figure, here GGGn corresponds to Gn or GGn for n= 1,2,3, all loops are removed and the information of "direction" in the edges are lost. This is the kind of nets one would find from the kind of text mining done in this thesis. (Please note that the interactions among proteins is not displayed in this figure.)

gene	description
agtr1	angiotensin ii receptor, type 1.
agtr2	angiotensin ii receptor, type 2.
apoa1	apolipoprotein a-i.
apoe	apolipoprotein e
cd36	cd36 antigen (collagen type i receptor, thrombospondin receptor).
ccr2	chemokine (c-c motif) receptor 2.
csf1r	colony stimulating factor 1 receptor, ...
icam1	intercellular adhesion molecule 1 (cd54), ...
ifngamma	interferon, gamma
mcp	membrane cofactor protein (cd46, ... antigen).
mcp1	microcephaly, primary autosomal recessive 1
mmp9	matrix metalloproteinase 9 (gelatinase b, ...)
nfkb1	nuclear factor of kappa light polypeptide... (p105)
pon1	paraoxonase 1.
sele	selectin e (endothelial adhesion molecule 1).
selp	selectin p (granule membrane protein 140kda, antigen cd62).
sra1	steroid receptor rna activator 1.
tgfb1	transforming growth factor, beta 1 (camurati-engelmann disease).
tgfb2	transforming growth factor, beta 2.
tnf	tumor necrosis factor (tnf superfamily, member 2).
tnfrsf5	tumor necrosis factor receptor superfamily, member 5 (under the synonyms cd40 and cd154.)
vcam1	vascular cell adhesion molecule 1.
vla4	very lateactivation antigen4 (vla4 is not included in the lists from HGNC)

Table 3.2: A list of genes that appear to have a larger relevance to atherosclerosis than random. This list was generated by reading the reviews [Hansson, Humphries and Morgan, Libby, Lusis] "by hand" and selecting all gene names that occur in at least two of them.

One could (and have) of course also construct protein-protein nets, metabolite-enzyme nets and so on.

3.8 What genes do we expect in atherosclerosis?

The reviews [Hansson, Humphries and Morgan, Libby, Lusis] were "read by hand". Genes occurring in at least two of them are displayed in table 3.2. These genes are considered to be of more than random relevance.

3.9 What is an urn?

The statisticians best friend is the urn [Blom]. In it we can put balls of different colors. If we draw a number of balls at random there is a well defined number of balls of a particular color we would expect and the variance of this number is also well defined.

Let us for example put blue and green balls in the urn and assume that the fraction

of blue balls in the urn is p and the fraction of green balls is q (with $p + q = 1$). If the total number of balls is M and we draw m balls at random the stochastic variable X(m) with the meaning "how many blue balls do we get if we draw m balls from the urn" has the expected value $E(X) = mp$ and the variance $V(X) = \frac{M-m}{M+1}mpq$. (The variance is the standard deviation squared.) X has the well known hypergeometric distribution.

Let us imagine an urn with 300 balls where 150 of them are blue. If we draw 200 at random from the urn we would expect $np = 200 \cdot \frac{150}{300} = 100$ blue balls and a variance of 16.61 so the standard deviation is thus 4.07. We call the scenario where we find as many blue balls as expected, the null hypothesis, or H_0 for short.

If we have a system somehow selecting the balls from the urn, and from 200 drawn balls find 75 blue ones we can calculate the distance from H_0 in standard deviations as $\frac{X-E}{\sqrt{V}}$. In this case we are $\frac{-25}{\sqrt{16.61}} \approx -6.13$ standard deviations from the null hypothesis. One often say that within an interval of plus/minus two standard deviations of the expected value 95% of the probability is distributed for the normal distribution. So this result is clearly lower than random.

If we instead consider the scenario of an urn with 3 million balls in which we still have 150 blue ones we find that $E \approx 0.01$ and $\sqrt{V} \approx 0.1$. So if we draw 200 balls and find 75 blue ones the distance from H_0 is $\frac{75-0.01}{0.1} \approx 750$ standard deviations. We are now clearly doing something better than random.

3.10 What is precision, recall and f-score?

It is often practical to count how many true or false positives and negatives a (new) model has compared to another (old and "true") model. A true positive, tp, is an item that is present (or true) in both models; a false positive, fp, is an item the new model proposes as present but that we know (by looking at the true model) is false; a true negative is an item that is false in both systems and a false negative is an item that should be present but that is not. Precision, recall and f-score use tp, fp, tn and fn to compute a number between zero and one (in the rest of this thesis we often display mF for milli F-score) where a number close to one indicates a match between the models.

Precision (P) has the definition: $P = \frac{tp}{tp+fp}$, with the intuitive meaning "how much of what I say is true is true".

Recall (R) has the definition: $R = \frac{tp}{tp+fn}$, with the intuitive meaning "how much of the truth do I find".

F-Score (F) is a kind of average of P and R: $F = 2\frac{PR}{P+R} = \frac{tp}{tp+fp+fn}$.

For more about P, R and F consult for example [Shatkay and Feldman]. As an example we can consider the two following scenarios: First let us imagine tthat we have a system of 300 documents and of these 150 that deals with the gene "gremlin", analogous to the urn in the previous section. If we apply a search engine giving us 200 documents out of which 75 deals with the gene we get: $P = \frac{75}{200} = 0.375$, $R = \frac{75}{150} = 0.5$, and $F = \frac{2 \cdot 0.375 \cdot 0.5}{0.375+0.5} = 0.42$. Now, if we consider a similar system but instead of having 300 documents we have 3 million documents. Let us also assume that there are still 150 documents dealing with the gene "gremlin". If a scan for the gene "gremlin" gave us the same 200 documents as above P, R and F would be the same.

Compared to the urn in the previous section there is an interesting difference here, the urn rewards something that is better than random but P, R and F does not. But on

the other hand P still tell us us the fraction of of what we have found that is true and R still tell us how much of the truth we have found.

McCoy: Shouldn't you be working on your time warp calculations, Mr Spock?
Spock: I am. (Resumes staring into space)

Star Trek

4 Methods

"AUTOMATED SYSTEM" is a concept that occurs very often in this thesis since one of the goals of this thesis is to create one. The concept "automated system" is very vague, almost generic, and we have chosen this phrase since it implies nothing more than it should. Of course an "automated system" could be implemented in a number of different ways and in this chapter we explain some of the details of the implementation used, many details are left out and the reader may consult the tutorial and the appendix for them.

4.1 The workflow

Abstracts were manually downloaded in big files, with tens or hundreds of thousands of entries per file, and stored as plain text. A program implemented in the free programming language Python (www.python.org) was used to extract information like title, MeSH terms and so on from the abstracts. The program then stored this information in a MySQL database (www.mysql.com). An example of python code is illustrated in figure 4.1. In this example we illustrate that the internal representation of the nets are hash tables - known for fast accessing - where an edge is a sorted tuple, for example the edge between mmp8 and apoe is the *tuple* (apoe,mmp8) since apoe is "smaller" than mmp8 alphabetically. This also clearly shows the syntactic superiority of python, it is really easy to read and understand the code! Figure 4.2 show an example of an interaction with MySQL when done "by hand". This was generally not done since a textual interface was created in python that communicated with both the user and the MySQL database. Figure 4.3 demonstrates how the interface looks and indicates some of the features included in the program.

Some of the more important algorithms are displayed in the appendix and some improvements are suggested. There is also an example of what type of output we can get in the end of this chapter.

4.2 The cardiovascular disease case

From PubMed we downloaded about 2 million abstracts. The abstracts were selected using a simple search for "cardiovascular disease" and related subjects in the standard search field in PubMed. The abstracts were initially stored as plain text in the semi-structured format in PubMed called "medline" as described in figure 3.1. The fields that were actually used was only the pmid, the title, the text in the abstract the MeSH terms.

```
def random_network(n,e):
    # function creating a random network (very similar to ER)
    # adding e edges

    if not ( (e < n*(n-1)/2) and (e > 0) and (n > 1) ):
        return {}

    net = {}

    while e > 0:
        a = randint(1,n)
        b = randint(1,n)
        while b == a:
            b = randint(1,n)

        (a,b) = (min(a,b),max(b,a))

        if not net.has_key((a,b)):
            net[(a,b)] = 1
            e -= 1

    return net
```

Figure 4.1: An example of python code. This particular function creates a random net with n nodes and e edges. If we try to add too many edges the program returns an empty net. The variables a and b correspond to the numbers of the edge (a,b) we try to add. The function *randint(x,y)* returns a random integer between x and y. This program is particularly suboptimal for very dense nets.

```
mysql> SELECT hgncid, gene, longname
    -> FROM gene
    -> WHERE longname like "beta%";
+--------+--------+-------------------------------------------------------------+
| hgncid | gene   | longname                                                    |
+--------+--------+-------------------------------------------------------------+
|    914 | b2m    | beta2microglobulin                                          |
|    915 | b2mr   | beta2microglobulin regulator                                |
|    921 | b3gat1 | beta1,3glucuronyltransferase 1 (glucuronosyltransferase p)  |
|    922 | b3gat2 | beta1,3glucuronyltransferase 2 (glucuronosyltransferase s)  |
|    923 | b3gat3 | beta1,3glucuronyltransferase 3 (glucuronosyltransferase i)  |
|    933 | bace1  | betasite appcleaving enzyme 1                               |
|    934 | bace2  | betasite appcleaving enzyme 2                               |
|   1047 | bhmt   | betainehomocysteine methyltransferase                       |
|   1048 | bhmt2  | betainehomocysteine methyltransferase 2                     |
|   1121 | btc    | betacellulin                                                |
|   1144 | btrc   | betatransducin repeat containing                            |
|  13815 | bcmo1  | betacarotene 15,15monooxygenase 1                           |
|  18503 | bcdo2  | betacarotene dioxygenase 2                                  |
+--------+--------+-------------------------------------------------------------+
13 rows in set (0.06 sec)
```

Figure 4.2: An example of SQL code demonstrating how SQL can work and what kind of output we get when using it “by hand”.

```
   help  - print a helptext.

   mesh  - get (nothing but) a list of mesh terms.
   addm  - add a mesh term (and its offspring) to the search-set.
   delm  - set mesh term (with offspring) not to be in set-definition.
   kids  - display selected mesh terms and their kids/offspring.
   abs   - count abstracts given the mesh terms.

   clear - delete all lists of mesh terms and start over.

   doit  - analyse the abstracts with selected mesh terms, producing....
             - raw output in html-format
             - minimal spanning tree using mesh terms as observations
             - minimal spanning tree using genes as observations
             - plot of the hierarchical cluster of mesh terms
             - plot of the hierarchical cluster of genes
             - plot of in/out-degrees of nodes
             - a data-file for generating a Bayesian network
             - more ?

   exit  - leave the program.


 * please feed me: doit
   Analysis-options...
      0 - default (human with prints)
      1 - turn all prints off and use human genes
      2 - yeast
      3 - yeast without prints
   What do you want? 0
 - the timestamp for this scan is 2005jan17_135416,
       2531 abstracts connected to the term: denmark
        308 abstracts connected to the term: scandinavia
       4780 abstracts connected to the term: sweden
 - potentially 7619 unique abstracts, total of 7415 unique ones.

...

 - completed: 2005jan17_140014, browse it in mozilla? <y|n>
```

Figure 4.3: Selected output from the interface of the program implemented. Here we have selected the MeSH term “scandinavia” and stopped the MeSH term “norway” and its child “svalbard” (not shown). We can also see that each scan got a time stamp as unique id. The scan took about 6 minutes and included 7415 abstracts.

Previous research has noted great problems with case variation for gene names so all abstracts were filtered into lower case only - all upper case letters were lowered. In the same spirit all hyphens or minus signs were deleted. One might argue that conflicts must arise from this and of course they do: the genes “trav11” and “trav1-1” for example were no longer “different” genes. This problem and some other problems were however ignored - all conflicting gene names were given the status “conflict” and were not used.

To find as many occurrences of genes as possible we used synonym lists. One big synonym list including for example old and withdrawn synonyms comes from [HGNC]. We used this list and considered an occurrence of any synonym to be an occurance of the “best” synonym (in the manner proposed by HGNC) of this synonym.

A large problem arises from research where authors are using different synonyms for the same gene. We would like to illustrate this problem with an example: the gene often referred to as “cd36” has the five synonyms “gp4”, “gpiv”, “scarb3”, “fat” and “gp3b”. Out of these six names for the same gene at least one, “fat”, is also a common word for something not being a gene. To filter these bad words out in a automated manner we used the dictionary from open office (www.openoffice.org) as a stop list. All the gene synonyms or gene names that are also words were filtered to avoid false positives. Some of these filtered genes are: “beta”, “grail”, “for”, “nude”, “cryptic”, “gremlin” and “dance”.

Another problem is for example the gene sialoadhesin, a macrophage adhesion molecule, that has the synonym “sn”. The word “sn” is also the short form of tin, element 50 in the periodic system. To avoid this kind of confusion we use the constraint on synonyms and gene names that they had to be at least three characters long, thus ignoring all shorter synonyms. Fortunately most genes with short synonyms also had longer ones: sialoadhesin, for example, has the longer synonyms “siglec1” and “cd169”.

Another set of bad gene names ignored was names without any letter such as the synonym “16.2” for immunoglobulin lambdalike polypeptide 2. Fortunately this gene also has the names “igll2” and “flambda1”.

There is of course the risk of a naming conflict - polysemy. Names like “beta3”, that is a synonym of the genes “eef1b3” (eukaryotic translation elongation factor 1 beta 3) and “bhlhb5” (basic helixloophelix domain containing, class b, 5), had to be deleted along with all other cases of polysemy. Due to the elimination of hyphens this category probably grew a bit.

We feel obliged to warn for problematic gene names that were not rejected but that might be unsuitable for the scientific community. Names like “smurf1” for “smad specific e3 ubiquitin protein ligase 1”, should be fought, but since there is no trivial way to make an automated system classing genes as “silly” or not these genes were unfortunately included. The groups of problem and how many names that belong to each group is described in table 4.1.

As briefly mentioned, the abstract meta-information in form of MeSH terms were stored in the database. We used these MeSH terms to select a portion of the abstracts by selecting a set of MeSH terms and if an abstract has this MeSH term, we included the abstract in the scan. Each MeSH term was also considered to “include” all its offspring (child-terms and grandchild-terms and so on) with the option of turning branches off. So if we were to select the MeSH term heart we would get all 23 offspring terms.[1] The

[1]The 23 terms are aortic valve, atrial appendage, atrioventricular node, bundle of his, chordae tendineae, ductus arteriosus, endocardium, fetal heart, heart, heart atria, heart conduction system, heart septum, heart

status	entries	example
nochar	7	47.11 - killer cell immunoglobulinlike receptor...
short	228	aa - atrophia areata, peripapillary chorioretinal degeneration
indict	555	pigs - phosphatidylinositol glycan, class s
conflict	1606	pit1 - solute carrier family 20... or pou domain, class 1...
ok	41725	apoe - apolipoprotein e

Table 4.1: A list of the status of the synonyms used in this thesis. About 40000 synonyms were downloaded from [HGNC] and classed as described above. Most were fine (ok) and used in the search for cooccurrences. Synonyms without any letter (nochar) were not trusted since any numerical result might be confused for a gene. Some were considered too short (short) and was not included in any search. If a synonym occured in a dictionary (indict) it was not trusted nor was cases of polysemy (conflict).

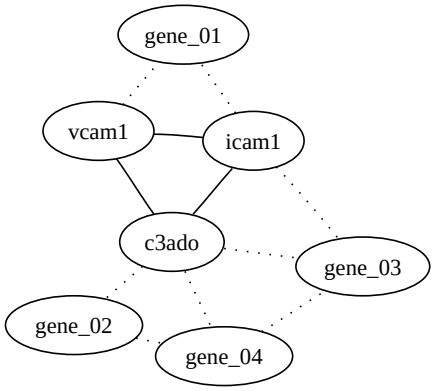

Figure 4.4: The full lines show what we would get from a scan of only the abstract in figure 3.1, please note that the algorithm would not find apoe since it is not a free word. The dotted lines indicate a possible net we could have found if we would scan more genes.

union of the abstracts linked to by the set of MeSH terms would then be used for a scan. We found 239000 pointers to abstracts and 177000 unique abstracts in this heart example. In these abstracts we now look for cooccurrences (as defined in the tutorial). In figure 4.4 we can see the gene network we might get if we scan only one or a few abstracts.

It is also convenient to use the same representation as one might use to describe gene expression matrices, as described in table 4.2. In the data set described by table 4.2 there are at least three cooccurrences in abstract1: gene1-gene3, gene1-geneN_1 and gene3-geneN_1.

There is also another way of replicating a gene expression matrix: instead of using one abstract per row we collapsed the abstracts referred to by a certain MeSH term so that there is a MeSH term per row instead. See table 4.2 for an illustration.

As the title of this thesis suggests the most important output from this thesis is the

valves, heart ventricles, mitral valve, myocardium, papillary muscles, pericardium, pulmonary valve, purkinje fibers, sinoatrial node, tricuspid valve and truncus arteriosus.

	gene1	gene2	gene3	gene4	...	geneN_1
abstract1	1	0	1	0	...	1
abstract2	0	0	1	0	...	0
⋮	⋮	⋮	⋮	⋮	⋱	⋮
abstractN_2	0	0	0	0	...	0

	MeSH1	MeSH2	MeSH3	MeSH4	...	MeSHN_3
abstract1	0	0	1	1	...	1
abstract2	1	1	1	0	...	0
⋮	⋮	⋮	⋮	⋮	⋱	⋮
abstractN_2	0	0	1	1	...	1

	gene1	gene2	gene3	gene4	...	geneN_1
MeSH1	321	15	5	0	...	1
MeSH2	1065	19	0	1	...	1
⋮	⋮	⋮	⋮	⋮	⋱	⋮
MeSHN_3	125	20	1	0	...	0

Table 4.2: An illustration of how an abstract to gene matrix (upper), how an abstract to MeSH (middle) and how a MeSH to gene matrix (lower) could look. In the abstract to gene matrix a 1 in position (i,j) is interpreted as "abstract$_i$ contains gene$_j$". A 1 in position (i,j) in the abstract to MeSH matrix has almost the same meaning as in the abstract to gene matrix: "abstract$_i$ is tagged with the MeSH term MeSH$_j$". These kind of matrices are typically very sparse and it is often a good idea to store only the nonzero elements. By a combination of the two upper matrices the lower is produced. In the MeSH to gene matrix a n in position (i,j) is interpreted as "out of all abstracts with the MeSH term MeSH$_i$ there are n dealing with the gene gene$_j$".

gene networks distilled from the abstract to gene matrix for a given abstract set. This gene network is believed to reflect the function described in the abstracts. If we were to use the MeSH term “heart” and all of its offspring we would hopefully get a gene net describing important genes functioning in the heart and heart-related diseases.

Of course, many things could be done with the “abstract to gene” matrix, among other things hit lists. In the hit list each gene got a score corresponding to how many abstracts it cooccurred in.

4.3 The yeast case

As for the cardiovascular case, abstracts were downloaded from PubMed. Now two sets of abstracts were downloaded; one set intended to target “cell cycle” and abstracts were selected with keywords like “cell cycle”; a second set was intended to target “saccharomyces cerevisiae” where abstracts were selected by using keywords like “yeast”. The two sets contained about 60000 and 220000 abstracts.

Gene and synonym lists were found from [SGD] with some additional synonyms from [Euroscarf]. The abstracts and synonym lists were treated in the same way as in the cardiovascular case with the exception that MeSH terms were not used to select subsets, instead the two cases were considered to be the only two possibilities. There was a large overlap of abstracts, half of the cooccurrence containing abstracts for the cell cycle case were also abstracts found in the yeast case.

For yeast the only output considered was the gene networks. As for the cardiovascular case, an edge had a label corresponding to the number of abstracts the nodes sharing it cooccurred in.

4.4 Example of a scan using the MeSH term foam cells

Via the interface implemented it is easy to select MeSH terms to use for a scan. One should note however that the results are always heavily biased towards cardiovascular disease due to the nature of our selection of abstracts.

In this example we selected the MeSH term “foam cells”, a MeSH term without children connected to 2467 abstracts. The raw output is in html, where one easily can click on genes to see with which other genes it is connected. There are two hit lists: the strongest cooccurrences (table 4.3) and and the most cooccurring genes (table 4.4). Small extra analyses are also conducted, such as the generation of a link distribution plot in figure 4.5. The main result of this example is shown in figure 4.6 where the net is shown.

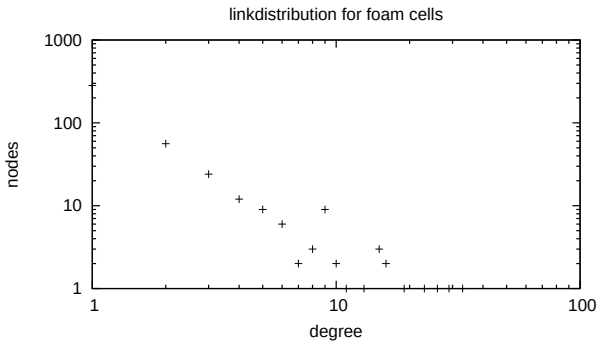

Figure 4.5: A link distribution plot for a scan using the MeSH term foam cells using almost 2500 abstracts. One can almost see a straight line (see next chapter).

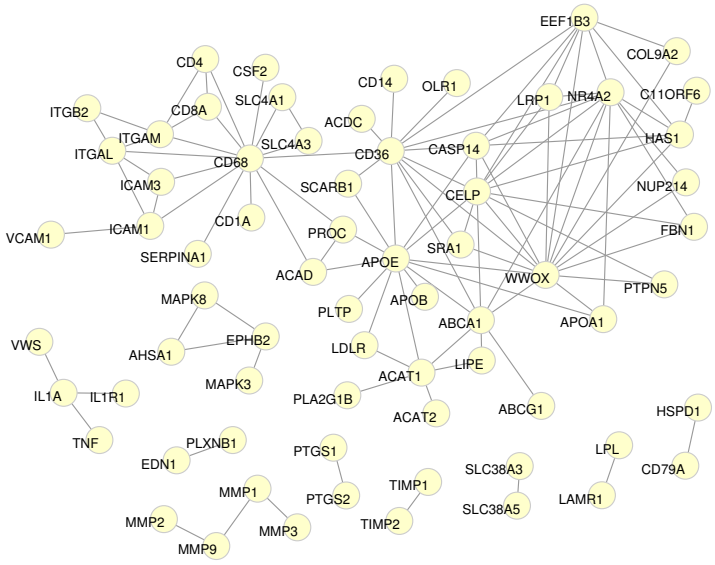

Figure 4.6: Having a cutoff at one (motivated in the next chapter) produces this net with about 69 genes and 109 edges. We get the indication that the following genes are important: wwox, cd68, nr4a2, eef1b3, cd36 and apoe, since they are hub-like.

strength	gene	gene	strength	gene	gene
14	celp	wwox	5	cd36	celp
14	acad	proc	5	acat1	apoe
11	nr4a2	wwox	5	lrp1	nr4a2
9	has1	wwox	5	casp14	eef1b3
8	casp14	wwox	5	ptgs1	ptgs2
8	eef1b3	wwox	4	celp	has1
8	cd36	wwox	4	abca1	apoe
8	cd36	sra1	4	celp	eef1b3
7	cd4	cd8a	4	mmp2	mmp9
6	lrp1	wwox	4	abca1	cd36
6	celp	nr4a2	4	casp14	celp
5	icam1	vcam1			

Table 4.3: Top cooccurrences for a scan of foam cells.

strength	gene	description
33	cd36	cd36 antigen (collagen type i receptor, thrombospondin…
29	wwox	ww domain containing oxidoreductase.
26	apoe	apolipoprotein e.
23	cd68	cd68 antigen.
19	acat1	acetyl-coenzyme a acetyltransferase 1 …
16	abca1	atp-binding cassette, sub-family a (abc1), member 1.
16	celp	carboxyl ester lipase pseudogene.
15	proc	protein c (inactivator of coagulation factors va and viiia).
15	acad	entry withdrawn.
15	nr4a2	nuclear receptor subfamily 4, group a, member 2.
13	vws	van der woude syndrome.
11	has1	hyaluronan synthase 1.
10	ldlr	low density lipoprotein receptor…
10	icam1	intercellular adhesion molecule 1 (cd54)…

Table 4.4: Top coccurring genes after a scan of foam cells.

If you're gonna compare a Hanzo sword, you compare it to every sword ever made...

Kill Bill vol. 2.

5 Validation and results

JUST AS IN MANY TV-SHOWS for children, such as "Sesame street" or "Fem myror är fler än fyra elefanter"[1], where two funny persons compare two or more items, there are many ways to compare graphs. One could almost say that the scientific community have not taken this issue seriously and with some kind of arbitrary hand-waving arguments use the validation method that suits them best. For *graphs* there are also many ways to validate if it is a good or bad model of "the truth". But we are convinced that the best way to do this must be to compare a *graph* describing a system to another *graph* describing the same system. If the results are more similar than one could expect from random we have a reason to trust the new results, or at least to assume that the new results model the same thing the old results model.

This chapter will begin with a short discussion of the link distribution of the nets. Then three sections discussing pairwise comparisons of graphs using different methods will follow. The third of these is new and has a great potential. All of these methods are used under the assumption that overlapping edges is the most important feature of the graphs.

A section investigating the use of a possible cutoff of the edges is included. We also compare lists of genes generated with different methods for the atherosclerosis case where there are no available nets.

5.1 Link distribution

A method used to characterize graphs is to look at their link distribution. The link distribution of a graph is a figure showing the number of nodes having a certain degree. Figure 5.1 is an example of this and shows a few examples. These plots are often drawn in log-log-scale since there, for many kinds of graphs, is an appearance of a straight line in this scale. This implies that the number of nodes having a certain degree is exponentially decreasing for increasing degrees. Many naturally occurring nets, and nets created by humanity, like food webs, the internet, metabolic nets and cooccurrence nets have this characteristic also known as a power law. See for example [Barabási] for a comprehensive discussion of this topic. We would of course expect the nets generated in this thesis to have this property. In an ER network (as described in the tutorial) this property is absent.

Gene networks generated in thesis followed this power law characteristic and had a slope of about -1.5 in log-log-scale, see figure 5.1.

[1] See for example http://www.svt.se/myror/myror2/container.html where one can compare: a snail, a turtle, an ostrich and a centipede.

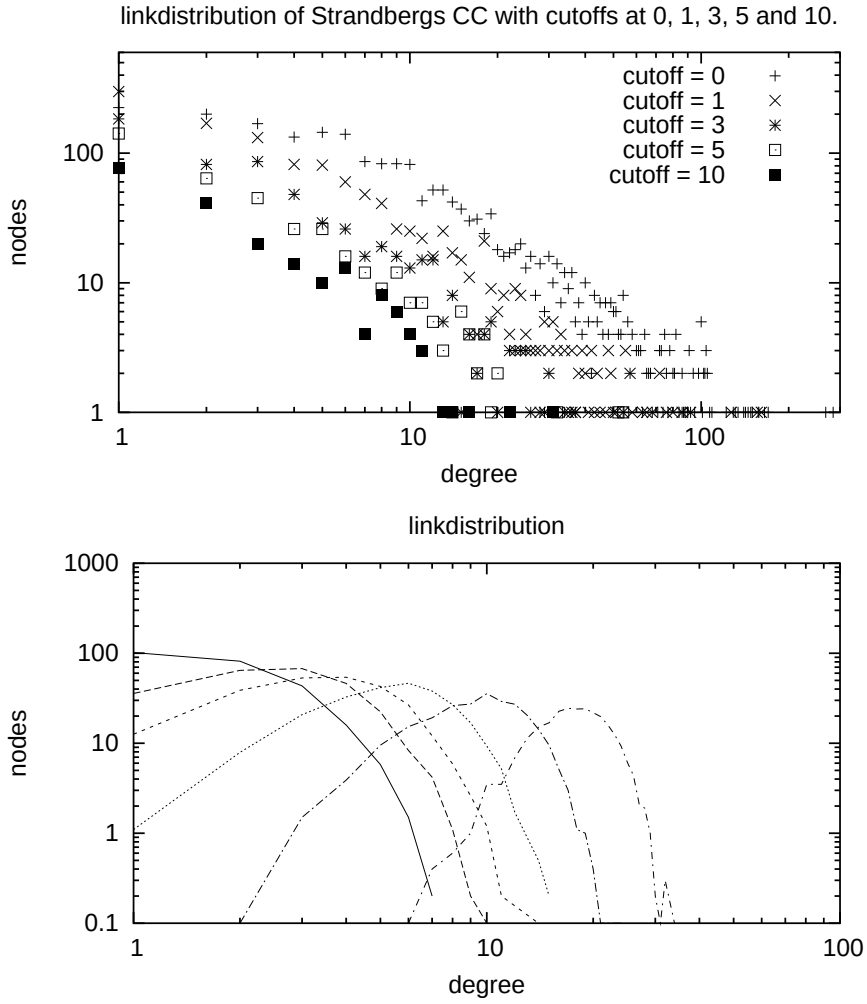

Figure 5.1: Link distribution for two kinds of graphs. The upper figure shows many link distribution scatterplots of a text mined net from abstracts dealing with “cell cycle” for yeast with cutoffs at 0, 1, 3, 5 and 10. Edges weaker than the cutoff are removed. One can suspect that the deviation from a straight line for a cutoff at zero could appear as an effect of unwanted connections among nodes. The slopes were -1.38, -1.45, -1.47, -1.42 and -1.50 respectively. The lower figure displays the link distribution from a net starting as a tree with 0, 128, 256, 512, 1024 and 2048 randomly added edges. The distribution tends to move towards less leaves and a peak at some non-zero mean. Each curve is a mean of ten networks. These curves do not resemble straight lines.

5.2 Pairwise comparisons of graphs 1: Overlap

The first method used to perform pairwise comparisons of graphs in this thesis is the simple way to just compute the number of overlapping edges, the true positives. An edge counts as overlapping if there, in both nets, is an edge between the same pair of genes. Here we do not consider the weights of the edges, they are all equally important. The uppermost part of table 5.1 show the number of overlapping edges for all possible pairs of nets using the cell cycle and Saccaromyces Cerevisae net from this thesis (as explained earlier), Luscombes cell cycle net, Lees net, Guelzims net, and the Insuk Lee. Insuk Lee has merged nets from different sources to get a net and Luscombe, Lee and Guelzim has used microbiological data to get their nets. It is difficult to say anything from these numbers, for example: our yeast net has about 560 overlapping edges when compared to the Guelzim net and Luscombes cell cycle net has about 520 overlapping edges when compared to the Lee net. Does this mean that our is a better model of the Guelzim net than the Luscombe net is a model of the Lee net? Well, one have to understand that our yeast net contains 41000 edges, Guelzims about 900, Luscombes about 800 and Lees net has about 4300 edges. In almost any method used to compare nets the number of overlapping edges is important. But we urge the reader to not interpret these numbers, we will let the reader see what for example P, R and F does with them.

5.3 Pairwise comparisons of graphs 2: P, R and F.

Instead of looking at just overlap, we might want to get a score to be able to say if the overlap is “good” and “bad”. F is displayed in table 5.1 and we remember that the rule of thumb is that if P and R (and thus F) are both better than 0.6 we have a nice result. Compared to this our results are awful. A paradox is that this does not imply that our results are worse than random, or even bad, in fact we shall see that the results are actually pretty good.

If we look at F, one can see that a bigger number of overlapping edges does not have to mean a better F. In table 5.1 one can for example see that our yeast net versus Guelzims net has a higher overlap but a lower F (overlap=558 and mF=26) than the Luscombes cell cycle net versus Lees net has (overlap=522 and mF=202). This example shows that better overlap does not have to mean better F.

5.4 Pairwise comparisons of graphs 3: the urn approximation

We must now remind the reader of the urn: If we have an urn with M balls, with the fraction p of them being blue, the fraction q of them being green and p+q=1 we can define the stochastic variable X(m) as the number of blue balls drawn from the urn if we draw m balls at random without replacing any of them. The expected number of blue balls is E(X)=mp with the variance V(X)=$\frac{M-m}{M-1}$mpq.

First we will now consider the case where we compare two *trees* (G_A and G_B) with the same nodeset (N) having n nodes. As an approximation we will use the metaphor of an urn, in this urn we will put as many blue balls as there are edges in G_A thus as many as there are nodes in N minus one since we are dealing with trees. We also add a number of green balls - one for each possible edge that is not present in G_A. We now

(i) Overlap	S CC	S SC	L CC	Lee	Gue	I Lee
Strandberg CC	15821	13845	103	100	158	2595
Strandberg SC	13845	41366	174	221	558	5228
Luscombe CC	103	174	807	522	142	48
Lee	100	221	522	4356	119	67
Guelzim	158	558	142	119	904	172
I Lee	2595	5228	48	67	172	54700
S CC with cutoff	4796	4181	67	52	95	1556
S SC with cutoff	4181	13439	112	130	393	3673

(ii) mF	S CC	S SC	L CC	Lee	Gue	I Lee
Strandberg CC	1000	484	12	9	18	73
Strandberg SC	484	1000	8	9	26	108
Luscombe CC	12	8	1000	202	165	1
Lee	9	9	202	1000	45	2
Guelzim	18	26	165	45	1000	6
I Lee	73	108	1	2	6	1000
S CC with cutoff	1000	458	23	11	33	52
S SC with cutoff	458	1000	15	14	54	107

(iii) Distance from H_0	S CC	S SC	L CC	Lee	Gue	I Lee
Strandberg CC	4241	2292	121	49	176	367
Strandberg SC	2292	4241	126	66	386	456
Luscombe CC	121	126	4241	1180	704	29
Lee	49	66	1180	4241	253	14
Guelzim	176	386	704	253	4241	102
I Lee	367	456	29	14	102	4241
S CC with cutoff	4241	2207	143	47	193	404
S SC with cutoff	2207	4241	143	70	477	569

Table 5.1: The six nets in these tables are compared to each other where each row is considered to be a model of each column. The three tables are displaying different validation techniques: (i): the number of overlapping edges in pairwise comparisons of graphs. In the nets all loops were removed and all edges are considered to be undirected. (ii): milli F-score (mF) for pairwise comparison of nets. (iii): The distance in sigmas from the null hypothesis. At first it might seem that this score is the same thing as F, but scaled, but this is however not true. The tables all have two additional lines where we have written what happened when a cutoff at 1 was applied. Interesting observations from the two additional lines tables are the fact that the distance increases in almost all cases when applying a cutoff, but that F does not seem to increase as clearly. A conclusion is that it must be a sad day for F when we find something one hundred standard deviations better than something else when F remains the same (compare our yeast net with the I Lee net with and without a cutoff). The urn score is generated under the assumption that there is a total of 6000 genes in the yeast genome.

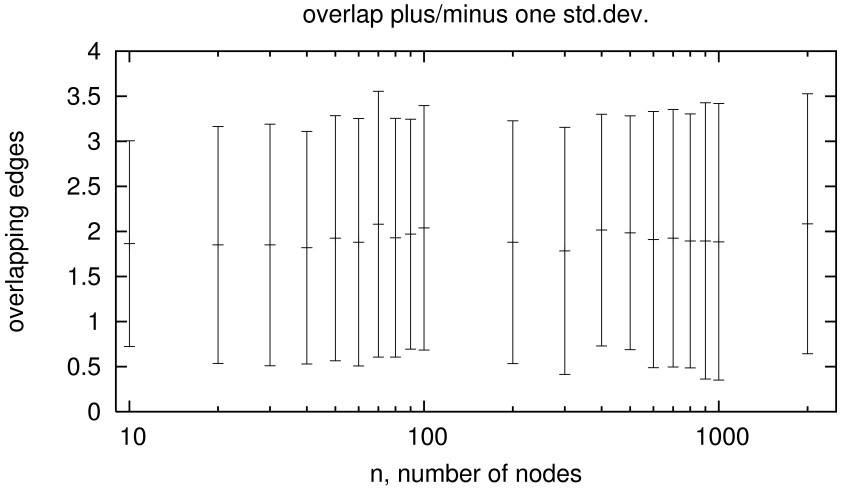

Figure 5.2: Random Overlap. The figure displays the mean number of overlapping edges plus/minus one standard deviation from 200 pairwise comparisons of random trees for various sizes. We are always close to $mean \pm sigma = 2 \pm \sqrt{2}$.

have as many balls in the urn as there are possible edges for N. We can think of each blue ball as an edge in G_A and each green ball as an edge that could have been in G_A but that is not there.

If we assume that the two trees are unrelated we could think of the second tree, G_B, as having edges drawn at random from the urn. This would correspond to drawing as many balls from the urn as we have edges in E_B. By calculating the number of blue balls we could expect if this was a random event, E(X), we can compare this number to how many blue balls we actually got: the number of overlapping edges. We know that X has a hypergeometrical distribution since we do not return any balls. Given what we already know of this distribution we notice that the expected value E(X)=$2\frac{n-1}{n}$ and that the variance V(X)=$2\frac{(n-1)^3(n-2)}{(n^2-n-2)n^2}$. We come to the remarkable conclusion that, for large values of n, we expect an overlap of 2 edges and a variance of 2. (If this observation is unnamed, we propose "Strandbergs number two".)

As an illustration of this assumption we did 200 pairs of random trees with a fix number of nodes and counted X, the overlapping edges, and sigma the standard deviation. This procedure was repeated for trees with 10 to 2000 nodes, see figure 5.2, and the mean value was very close to two no matter the number of nodes. The standard deviation was also close to the square root of two. One could expect that the urn would react in a different way when we compare two trees, since the trees have constraints. One constraint is that once some of the edges are added to what is to become the tree some edges are prohibited since we do not allow loops. The urn does not "feel" this constraint but apparently it does not matter.

We have just seen that this urn approximation seems to be valid for random trees, let us now assume that we are interested in a comparison of two networks. We can consider the pair of nets to be a pair of random trees with n nodes, saturated with k and r extra edges respectively. Still using the metaphor of the urn we now have $(\mathrm{n}-1)+\mathrm{k}$ blue balls, $\frac{n(n-1)}{2}-(\mathrm{n}-1)-\mathrm{k}$ green balls and we draw $(\mathrm{n}-1)+\mathrm{r}$ balls at random from the

	10	100	1000	10000
100 simu	2.94 ± 1.59	3.86 ± 1.79	13.8 ± 3.20	114 ± 8.87
100 calc	2.91 ± 1.70	3.92 ± 1.97	14.0 ± 3.67	115 ± 8.82

Table 5.2: To a pair of trees with 250 nodes 100 edges was added to one of them, to the other 10, 100, 1000 or 10000 edges was added. This procedure was repeated 100 times for each combination of extra edges and the mean overlap and standard deviation was noted. In the row labelled "100 simu" the result from this simulation is noted in the form $mean \pm sigma$. The row "100 calc" describes the expected overlap and standard deviation we would get with the urn approximation. In 15 cases of 16 the analytical mean fitted into a 95% confidence interval around the simulated value.

urn. We now expect E(X) $= 2\frac{(n-1+k)(n-1+r)}{n^2} \approx 2\frac{(n+k)(n+r)}{n^2}$ edges overlap for large values of n and a variation of V(X) $= 2\frac{(n-1+k)(n-1+r)}{n^2(n-1)^2}\frac{(n^2-3n+2-2k)(n^2-3n+2-2r)}{n^2-n-2} \approx 2\frac{(n+k)(n+r)(n^2-2k)(n^2-2r)}{n^6}$ for large values of n. As an example we now consider the expected overlap of two networks where we have three times as many edges as we have nodes (k $=$ r $=$ 2n), for example a yeast net with 6000 genes and 18000 edges: for large n's we would expect 18 edges overlap with a sigma of about 4.24. Here the reader must understand that an overlap of 18 would give poorer and poorer F the more we increase n.

Again we performed simulations to compare the urn-assumption with real networks. To 100 pairs of trees with 250 nodes, k edges were added to one of them, and r to the other one. We calculated the mean overlap and the variation. This was repeated for values of k and r at 10, 100, 1000 and 10000. Around the simulated mean a 95% confidence interval was made and in only one case the analytical mean ended up outside, so if we were to use the null hypothesis "the methods are different" we could not reject it. This does not have to mean that we have proven the validity of this approximation, but we have not rejected it. Again simulations and calculations seem to give pretty much the same results. As an example the simulation of the E- and sigma-values and the analytical values are described in table 5.2 for k = 100 and r at 10, 100, 1000 and 10000.

So far we have looked at trees and quasi ER graphs. Let us now investigate what happens if we focus on nets having a power law link distribution instead. It is not as easy to simulate a net with fixed link distribution having a well defined number of edges. So the nets used to do pairwise comparisons now had a well defined number of nodes and a varying number of edges.[2] Two sets of nets were made: the first set "sizevar" had a varying number of nodes and about three times as many edges as nodes. The second set "degvar" had 250 nodes and a varying number of edges.

Results from a comparison of the sizevar trees compared to quasi ER graphs showed that we are very close to the null hypothesis (between -0.35 and +0.04 sigmas from H_0), close enough to say that we do not have to modify the null hypothesis for these kinds of comparisons. The graphs of each size were also compared internally (all possible pairs of each size were compared) using the urn approximation. From these comparisons we noted that the results were similar to what we got when comparing with random graphs but that there is a slight tendency to find less information than what the null hypothesis claims. As for the sizevar set the degvar set was also compared to quasi ER graphs

[2] All these nets wery created by Björn Brinne, thank you Björn.

and compared internally, the results were similar to those for sizevar. As we hoped it is thus no big difference in results when comparing quasi ER graphs with graphs having preferential attachment.

We have seen pairwise comparisons of simulated nets of different classes. The results are promising: (i) for quasi ER random graphs the overlap is very close to what we would expect if we were to use this approximation. So close that we can say without to much of doubt that the approximation is a valid approximation. (ii) When we compare quasi ER random graphs with graphs having preferential attachment we are also very close to the null hypothesis. (iii) When comparing pairs of graphs having preferential attachment we are also pretty close to the null hypothesis but (iv) there seems to be a slight tendency of this approximation to expect a little more than simulations would expect us to find. Fortunately this will only suggest that if we find something far from the nullhypothesis (on the positive side) we might actually have done something more significant than we think.

Results from pairwise comparisons of some nets and the distance in sigmas from the nullhypothesis are displayed in table 5.1. The distance from the nullhypothesis is also studied in figure 5.3 as a function of cutoff, as described in the next section.

We are now ready to look at this score compared to overlap and F. Here we can now for the first time (?) see that the low values of overlap, P, R and F sometimes are in contrast to their intuitive interpretation: that a low value would mean that we are close to doing things at random. We can see that or yeast net versus Guelzims net (overlap=557 and mF=26) ends up 386 standard deviations from what we could expect if the nets were unrelated. Luscombes cell cycle net versus Lees net (overlap=522 and mF=202) has a distance from the null hypothesis of 1180 standard deviations. So we dare say that Luscombes cell cycle net and Lees net are “closer” than our net to Guezims net. This is what we would have expected since the nets, in their construction, are related. We might now tempted to say that “higher F means longer distance from the null hypothesis”, but this is not generally true: a counter example is the cell cycle and yeast nets compared to Luscombes cell cycle net, here the best distance is not the pair with the best F.

5.5 The cutoff

The strength of an edge in the text mining-graphs is the number of abstracts in which the gene pair has cooccurred. It is not unreasonable to assume that the higher the score the more relevant edge. One could argue and say that recent findings of the true gene interactions are not specified in an abstract since they are often performed in such large scale that the genelists have to be excluded from the paper. We will therefore not say that a high score means “more relevant edge” in a biological sense.

The opposite seems to be true however: we seem to be able to interpret really low scores as “less relevant edge” and there are two major motivations for this. The first is illustrated in figure 5.1 where one can see that the cooccurrence graph seem to have more of a power law characteristic for a non zero cutoff. This could be interpreted as follows: there are a number of false relations shortcutting the net and reconnecting the true leaves and thus lowering the number of observed leaves. This is consistent with results in [Jenssen et al] where a degree of false positives of 40% for edges of weight one versus 29% for edges of weight five or more was noted.

The other motivation of the hypothesis that low scores imply “less relevant edge” is shown in figure 5.3 where P, R, F and distance from the null hypothesis in sigmas

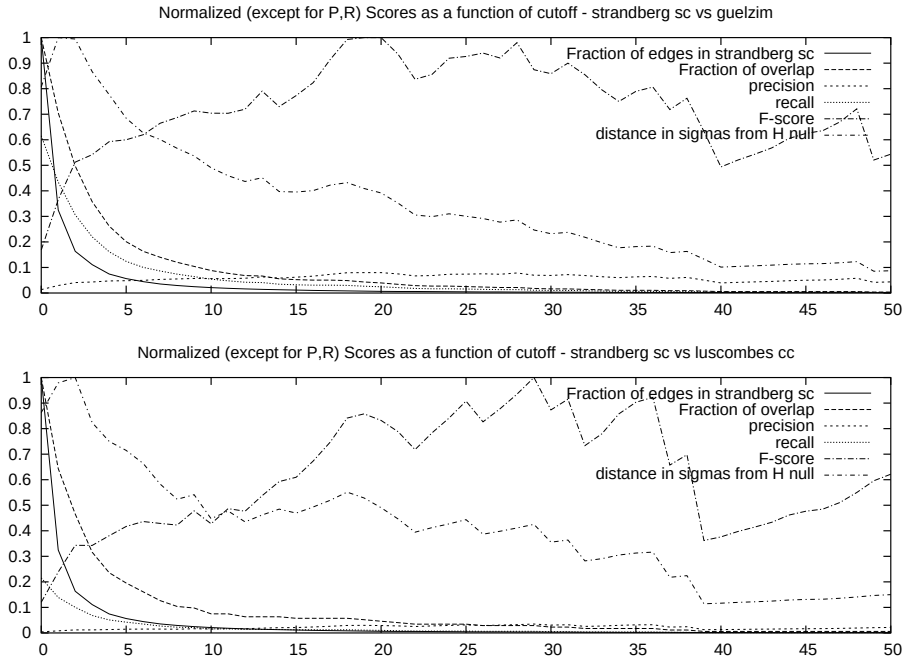

Figure 5.3: Two plots showing the fraction of edges left, the fraction of the overlap, P, R, F (normalized so that its maximal value is one) and distance in sigmas from the null hypothesis (also normalized) as functions of the cutoff. Things of importance in these plots are (i) the rapid decline in edges left and the not as rapid decline in the fraction of overlap left for low values of the cutoff; (ii) the peak of the distance in sigmas from the null hypothesis for a cutoff at 1, 2, or 3; (iv) the peak of F when the cutoff is between 15 and 40 and (v) P and R are extremely poor (not normalized in the plots); (v) The zig-zag-like pattern in both F and the distance in sigmas from H_0 could be interpreted as an effect from the low number of tp left: one tp more or less has a great impact. These observations support the hypothesis that there is an important portion of fp for cutoffs lower that 1, 2 or 3.

(among others) are plotted as a function of the cutoff. This figure shows that the distance from the null hypothesis in sigmas seems to have a peak for cutoffs around 1, 2 or 3, implying that there is a higher portion of fp in these weak edges. The same figure illustrates that F seems to be particularly large for a cutoff somewhere between 15 and 40, an interval where many edges are gone. This could possibly imply that there is a high degree of tp in this area. We therefore give the hint to the reader to use a cutoff at 1 in this case. This is motivated by three things: (i) about 60 to 70% of all edges are deleted but only 20 to 30% of the overlap is lost and (ii) we are as far away from a random event as possible. (iii) The straight line in the link distribution is clearer with a cutoff than without.

5.6 Comparing hit lists with hit lists

One of the more human readable forms of output are the hit lists. The hit lists generated here are constructed by giving a gene a score that is equal to the number of times it has cooccurred. We will in this section compare such a list, generated from a scan of 66000 abstracts with a total of 1367 genes, with lists generated from reading the reviews [Hansson, Humphries and Morgan, Libby, Lusis]. Each potential gene name was added to a list, so there were four lists from the reviews with 10 to 52 genes, where a mild filter had been applied. The union of the four lists was also generated (with 96 genes), and finally a list with the genes that occurred in at least two of the reviews. The text mined list was also used to generate two more lists: one with the top 134 genes (the top 10%), and another with the top 800 genes (genes with a score greater than one). One final list was gathered from the additional material of [Seo et al]. This last list is generated from gene expression experiments and contains 206 genes considered to be connected with atherosclerosis (where at least one had a name that was filtered).

Each possible pair of lists was compared by counting the overlap and by computing the distance from the null hypothesis in sigmas. Selected results are displayed in table 5.3. When we compared the lists some interesting results emerged. When the lists from the reviews were compared to our lists, the top 10% list always scored the best. But when compared to the list from [Seo et al] the best one was the full list. One possible interpretation of this is that the genes that score best in our hit list are well studied (and cooccur often) and are thus mentioned in the reviews whereas genes that are only studied a bit might actually be more important than the reviews imply - or are not as well understood and therefore not mentioned in the reviews.

So far, we have had a constraint of a perfect match, meaning that for example two genes in the same family (for example mmp1 and mmp2) does not match unless their numbers are correct (mmp1 does not match mmp2). To ease this constraint and count mismatches within a family as a hit, all numbers in the gene names were turned into a tilde (so that both mmp1 and mmp2 were both turned into mmp$\sim$ and now matched). Now there were 7600 “unique” names instead of 20000. Another interesting observation when comparing the now filtered lists is that the scores now were worse compared to the reviews but better compared to the list from gene expressiondata. A possible interpretation is that the reviews actually have the correct number after all and that the gene expression data does not.

A problem with this method is of course that genes differing in only a number might not be family, and genes within a family might not be separated in notation with just a number, like “apoe” and “apob”.

Out of the 72 genes that occurred in any review 18 were hit by the text mining algorithm. The 54 genes that were not hit were examined “by hand”, and at least 29 of them should have been hit by the algorithm. At least 15 of the genes were not found because of interpretation problems: when typing the genes from the reviews these genes were translated “by hand” to raw text and often the genes were written in the review without a definition of their abbreviation. It was not clear that for example “β” was to be interpreted as “beta” whereas it might have been “b” in the database - or the opposite. More problems are written in table 5.4.

The intersection of genes found from both text mining and in [Seo et al] is described in table 5.5 and in 5.6.

	Strandberg	Strandberg 2+	Strandberg 10%
[Humphries and Morgan] (10)	6.7 (6)	9.0 (6)	19 (5)
[Libby] (34)	5.2 (10)	7.6 (10)	14 (7)
[Hansson] (52)	4.6 (12)	7.0 (12)	13 (8)
[Lusis] (31)	4.2 (8)	6.2 (8)	15 (7)
Union (96)	8.3 (27)	12 (27)	22 (18)
Review 2+ (22)	3.8 (6)	5.6 (6)	15 (6)
[Seo et al] (205)	3.6 (27)	2.1 (14)	2.3 (4)
[Humphries and Morgan] (10)	4.2 (6)	5.8 (6)	12 (5)
[Libby] (28)	1.1 (6)	2.5 (6)	6.8 (5)
[Hansson] (36)	1.9 (9)	3.6 (9)	8.5 (5)
[Lusis] (27)	1.8 (7)	3.3 (7)	8.5 (7)
Union (72)	2.7 (18)	5.0 (18)	11 (6)
Review 2+ (19)	2.2 (6)	3.6 (6)	10 (13)
[Seo et al] (180)	6.2 (54)	5.9 (37)	4.8 (11)

Table 5.3: Some of the results originating from pairwise comparisons of lists. Each row correspond to a list originating from reviews [Hansson, Humphries and Morgan, Libby, Lusis], their union, the list of genes that occur in at least two of the reviews or from the additional material of [Seo et al]. The number of genes in each list is displayed within (). The three rightmost columns corresponds to lists generated from a scan of 66000 abstracts. In the upper half of the table no number filter is used, and in the lower half all numbers are filtered. The number displayed is the distance from the null hypothesis in sigmas and the absolute number of overlapping genes within (). The number of genes in our lists are for the upper half 1367, 800 and 134; and for the lower half 1073, 646 and 123. The reader should note that in both the upper and the lower halves the best distance is achieved by the top 10% list in all cases but one: when the lists are compared to a list originating from a gene expression experiment in [Seo et al] where the full list gave the best result.

Problem	Names	Example
Mea Culpa	15	eselektin
Greek or Roman	10	nfkappab and apoai
Family or antifamily	9	mmp and apob$\sim$
Word	2	age
Short	1	nf
In the database	29	hsp$\sim$

Table 5.4: The lists Review union and Strandberg full from table 5.3 were compared and the genes from review union that was not found in Strandberg full were manually investigated. The table categorizes come problems to why they were not found (one name may have more than one problem). The main problems were caused by us when typing the genes from the reviews, “mea culpa”, typically these genes were used without defining their abbreviation, or that we interpreted for example “β” as “beta” whereas it might have been “b” in the database - or the opposite. Another problem are genes within a family separated with roman (i, ii, iii, and so on) or greek (alfa, beta, and so on) letters or numbers, “greek or roman”. With “family or antifamily” we class genes that are unnumbered in the review but numbered in the database (family) or the opposite (antifamily). Two genes were named as if they were words (“age” and “mig”) and one was to short (“nf”). Of the 54 genes in the list about 29 were in the database and should have been found if they cooccurred in at least one abstract.

5.7 Validation and results: Recapitulation

The first section in this chapter dealt with the link distribution of the graphs. This section showed us that there is a power law link distribution in the graphs and that this is a little more obvious with a non zero cutoff. This is what we would have expected and what we hoped for.

The following three sections all dealt with overlapping edges since overlap, precision, recall, f-score and the urn approximation have this in common. The section that dealt with overlap is important since it shows us how little overlap there is. This was also illustrated in the section dealing with P, R and F: we saw how poor results we had using these standard tools. But instead of falling into despair we wanted to know why this was the case and how much overlap we would expect at random. By pure luck the first thing we did was the simulation of overlapping trees - Strandbergs number two - that is illustrated in figure 5.2. Of course we thought there was something wrong with the algorithm doing the comparisons so, again by pure luck, we had to do it analytically and found that we could not expect much overlap at random. Moreover we found *how* little overlap one would expect.

Again, these three sections have at least one thing in common: the idea that when comparing two graphs, you say that the graphs are more similar if they have more overlapping edges. This is of course a very reasonable idea and also a very simple way to measure similarity.

The sixth section of this chapter is especially important since it names the genes we might expect to be relevant (or at least studied in connection to cardiovascular disease). Comparing lists using absolute numbers is probably needed since we want to know what genes that overlap and therefore how many. Using an urn here is needed to illustrate that this is also very far from a random event.

gene	description
akr∼b∼	aldoketo reductase family ∼ member b∼ (aldose reductase)
apoe	apolipoprotein e
arhgap∼	rho gtpase activating protein ∼
arhgef∼	rho/rac/cdc42 guanine nucleotide exchange factor (gef) ∼
capg	capping protein (actin filament), gelsolinlike
cbx∼	chromobox homolog ∼
cdkn∼b	cyclindependent kinase inhibitor ∼b
cd∼	cd∼ antigen
chi∼l∼	chitinase ∼like ∼
cxcr∼	chemokine (cxc motif) receptor ∼
cxorf∼	chromosome x open reading frame ∼
c∼orf∼	chromosome ∼ open reading frame ∼
enpp∼	ectonucleotide pyrophosphatase/phosphodiesterase ∼
esd	esterase d/formylglutathione hydrolase
fbln∼	fibulin ∼
fbp∼	fructose1,6bisphosphatase ∼
fcgr∼a	fc fragment of igg, high/low affinity i/ii/iiia, receptor for (cd16/32/64)
fgfr∼	fibroblast growth factor receptor ∼
gstt∼	glutathione stransferase theta ∼
icam∼	intercellular adhesion molecule ∼
ifi∼	interferoninduced protein ∼ or interferon, alfa/gammainducible protein ∼
il∼ra	interleukin ∼ receptor, a/alpha
irak∼	interleukin1 receptorassociated kinase ∼
itgb∼	integrin, beta ∼
kiaa∼	kiaa∼
lamb∼	laminin, beta ∼
lta	lymphotoxin alpha (tnf superfamily, member 1)

Table 5.5: First half of the genes found by text mining of 66000 abstracts that also occur in [Seo et al]. Please note that the name c∼orf∼ corresponds to about has about 1200 different genes and that fcgr∼a can have high or low; i, ii or iii and is a receptor for cd16, cd32 or cd64.

gene	description
mapk∼	mitogenactivated protein kinase ∼
map∼	microtubuleassociated protein ∼
mmp∼	matrix metalloproteinase ∼
myh∼	myosin, heavy polypeptide ∼
nr∼h∼	nuclear receptor subfamily ∼, group h, member ∼
olr∼	oxidised low density lipoprotein (lectinlike) receptor ∼
plaur	plasminogen activator, urokinase receptor
ppp∼r∼b	protein phosphatase ∼, regulatory (inhibitor) subunit ∼a/b
psmb∼	proteasome (prosome, macropain) subunit, beta type, ∼
ptpn∼	protein tyrosine phosphatase, nonreceptor type ∼
rtn∼	reticulon ∼
runx∼	runtrelated transcription factor ∼
sca∼	spinocerebellar ataxia ∼
sdc∼	syndecan ∼
sds	serine dehydratase
slc∼a∼	solute carrier family ∼, member ∼
sod∼	superoxide dismutase ∼, soluble/mitochondrial/extracellular
spint∼	serine protease inhibitor, kunitz type ∼
spp∼	secreted phosphoprotein ∼
stat∼a	signal transducer and activator of transcription ∼a
stk∼	serine/threonine kinase ∼
tbx∼	tbox ∼
tlr∼	tolllike receptor ∼
tnc	tenascin c (hexabrachion)
tnfrsf∼b	tumor necrosis factor receptor superfamily, member ∼b
usp∼	ubiquitin specific protease ∼
znf∼	zinc finger protein ∼

Table 5.6: Second half of the genes found by text mining of 66000 abstracts that also occur in [Seo et al]. Please note that the name slc∼a∼ corresponds to hundreds of different genes and that sod∼ has one of soluble, mitochondrial or extracellular and not all.

Cogito cogito ergo cogito sum
(I think that I think,
therefore I think that I am.)

Ambrose Bierce, The Devil's Dictionary

6

Summary, conclusion and discussion

AS THE TITLE OF THIS THESIS INDICATES the goal of this thesis is to implement an automated system doing text mining to find gene nets for cardiovascular disease. In order to know if our nets are any good we have generated nets for yeast and validated them against other yeast gene nets.

Throughout this thesis we have been heavily inspired by the project described in [Jenssen et al] and when implementing the automated system that was to do text mining we wanted to solve some of the problems mentioned in it. One main improvement has to be the brutal treatment of gene and synonym names from the assumption that scientists do not know how to spell genes. By forcing the abstracts to be completely in lower case and by eliminating all minus signs we must have gotten rid of extremely many problems. But, as stated, some problems were created: the most obvious one is that different names might now be the same. To solve this we had to ignore some gene names. See table 4.1 for details of how many synonyms that had to be ignored. Another important improvement is that this system allows users to select a subset of PubMed by using MeSH terms in order to get the subnet of for example an organ or a disease.

As explained in the tutorial an edge in a cooccurrence graph corresponds to how often the gene pair has cooccurred. This integer is examined in the chapter dealing with validation and results. We can for example in figure 5.1 and figure 5.3 see that the cutoff is important. In the nets generated here we can see that a nonzero cutoff makes the power law link distribution clearer and that we move away from making a net at random if we have for example a cutoff at one instead of zero.

Recently, many gene nets have been generated using different methods. The best way to evaluate the nets generated in this thesis must, of course, be to compare them with other nets. The intuitive way to do this is to count the overlapping edges, but unfortunately there is hardly even an overlap of one percent. Another standard way of evaluating the nets is to look at P, R and F. The rule of thumb is that an F above 0.6 is good. Again the results are horrible, as best only about 0.1. The apparently poor results troubled us and we decided to find how much one could expect if one compared two random nets. We came to the remarkable conclusion that, for trees, we could only expect an overlap of two edges, no matter how big a tree one has! For two nets, both with n nodes, with n+k and n+r edges we would expect $E \approx 2\frac{(n+k)(n+r)}{n^2}$ overlapping edges and a variance of $V \approx 2\frac{(n+k)(n+r)(n^2-2k)(n^2-2r)}{n^6}$ for reasonably large values of n. Do note that k and r may be negative. If we found X overlapping edges we get a distance in sigmas from the null hypothesis of $\frac{X-E}{\sqrt{V}}$. This method uses an the approximation of an urn - something that makes it natural to use for ER graphs since

these graphs are very similar to using an urn. But since we do not compare pairs of ER graphs the we made careful simulations of different classes of graphs and found that there is practically no difference and that this approximation must be valid for at least ER graphs and graphs having preferential attachment.

By using this method and by looking at the link distribution for different cutoffs we found that the best cutoff is probably a cutoff eliminating all edges that only cooccur once. We now ended up between 47 and 569 standard deviations from what we would have expected if this was a random event. This is an incredible distance.

Of great importance is of course the genelists. These are lists of the genes that are studied in combination with the given MeSH terms. The case that interests us the most is when we selected the MeSH term arteriosclerosis and its offspring and got about 66000 abstracts. This scan resulted in a list including 1367 genes. From four reviews we, "by hand", created lists of genes that were mentioned, including 10, 34, 52 and 31 genes. The union of the genes occurring in these four review lists had 96, and a list that included all genes that occurred in at least two reviews had 22 genes. The final list considered is a list originating from gene expression data with 206 genes. When the review lists were compared to the top 134 genes in our list the overlap was between 13 and 19 standard deviations better than expected. The union list compared to the top 134 genes ended up 24 standard deviation from what we would have expected. We interpret these results, and the fact that it was the list with only the top genes that scored the best, as follows: if a gene is known to be connected to atherosclerosis someone will write a paper about it. If the gene ends up in many papers the probability that it ends up in a review increases. So if a gene is mentioned in many papers it (i): gets a higher score in the toplist and (ii): ends up in a review.

When our genelists were compared to the list originating from a gene expression study it was no longer the top genes that gave the best scores, nor was it the top 800 genes (all the genes that has cooccurred more than once). It was the full list that had the best score: 3.6 standard deviations from what we could have expected. This must be interpreted as the following: any gene that is relevant to cardiovascular disease has an elevated probability of ending up in a paper since it might be found to be connected to the disease. But only a few of these genes will end up in many papers and the ones that do are not always more relevant than the ones that do not.

In some cases the reviews mentioned genes with similar names, but that differed in only a number such as the members of the mmp-family. In at least one case the gene mmp3 occurred in one review and in another review five other mmp genes occurred (mmp1, mmp2, mmp8, mmp9 and mmp13). Of course a human reader must interpret this as a match since in at least two reviews by different authors the mmp-family occurs, but the automated system comparing the genes does not count this as a match since the word mmp3 is not the same word as mmp2! In order to count this situation as a match we eliminated all numbers that occurred in any gene name so that all members starting with mmp and ending with a number counts as the same gene. After this filter the hit lists got lower scores compared to the reviews but a higher score compared to the gene expression study. The lower score when compared to the reviews must be interpreted as an indication that the genes found actually matched, even with the numbers. The higher score, when compared to the gene expression study, is an indication of the opposite: the genes studied and named match the correct family but not as often the correct member.

The over all conclusion of this final thesis is that text mining of selected abstracts is a valid way to find a relevant gene net of an organ or disease. A discussion of the extension of this work is therefore motivated, one must continue to expand the use of textmining. To continue where this thesis ends is not hard, there are a number of issues

that have not been fully investigated. For one thing the use of filters in the abstracts: are the filters used relevant? Would we have found better results without the filters? Could we apply some other set of filters to improve the results? What happened if we start filtering authors? Journals? Dates? We have no clear answer to any of these questions but it is not unreasonable to assume that a filter, somehow taking care of roman numbers and the overused greek letters alfa, beta, gamma and so on could improve any future work. It could also be of valuable interest to instead of applying a very discrete cutoff use a more probabilistic approach, to assign a high or low probability of the word "gremlin" to be a gene or not. This whole thesis could be redone with some kind of probabilistic approach in every assumption used.

Of great importance is also the quite new concepts of integrating results from different fields into one model, as explained in [Seo et al]. Here a cooccurrence net could be a valuable complement if we are to use for example gene expression as another major component. Any edge proposed by both sources must be more trusted than an edge from only one.

One good thing with text mining is that it can produce a lot of edges in a gene net, but a drawback is that we cannot do novel findings by looking at only cooccurrences: the number of false negatives must be pretty elevated since we cannot find relations that have not been studied. Perhaps there is a way to look at the gene net and from it add edges in a controlled way by for example adding all possible edges in cluster like structures and in such a manner get a better idea of all possible positives.

As final words we must repeat that the over all conclusion of this thesis is that text mining of selected abstracts is a valid way to find a relevant gene net of an organ or disease.

Bibliography

[AHA] American Heart Assosiation, www.americanheart.org/
(Visited January 2005.)

[Barabási] AL Barabási. *The physics of the Web*. Physics World, **14, 33** (2001).

[Björn and Turesson] A Björn and BO Turesson. *Diskret Matematik*. Bokakademin, Linköping (2001).

[Blom] G Blom. *Sannolikhetsteori och statistikteori med tillämpningar*. Studentlitteratur, Lund (2002).

[Borgatti] SP Borgatti. *How to explain hierarchical clustering.* Connections **17** pp. 78-80 (1994).

[Euroscarf] Euroscarfs synonym list:
www.uni-frankfurt.de/fb15/mikro/euroscarf/syn_list.html
(Visited July 2004.)

[Guelzim et al] N Guelzim, S Bottani, P Bourgine and F Kepes. *Topological and causal structure of the yeast transcriptional regulatory network*. Nat Genet **31** pp. 60-3; PMID 11967534 (2002).

[Hansson] GK Hansson. *Immune mechanisms in atherosclerosis*. Arterioscler Thromb Vasc Biol. **21** pp. 1876-1890 (2001); PMID 11742859.

[HGNC] HGNC: www.gene.ucl.ac.uk/nomenclature
(Visited July 2004.)

HM Wain, MJ Lush, F Ducluzeau, VK Khodiyar and S Povey. *Genew: the Human Gene Nomenclature Database, 2004 updates*. Nucleic Acids Res. **32** (2004); Database issue:D255-7; PMID 14681406.

HM Wain, RC Lovering, EA Bruford, MJ Lush, MW Wright and S Povey. *Guidelines for Human Gene Nomenclature*. Genomics **79** (2002), pp. 464-470; DOI:10.1006; PMID 11944974.

[Humphries and Morgan] SE Humphries and L Morgan. *Genetic risk factors for stroke and carotid atherosclerosis: insights into pathophysiology from candidate gene approaches*. Lancet Neurol. **3** pp. 227-235 (2004). PMID 15039035.

[Jenssen et al] TK Jenssen, A Laegreid, J Komorowski and E Hovig. *A literature network of human genes for high-throughput analysis of gene expression*. Nat Genet. **28** pp. 21-28 (2001). PMID 11326270.

[Lee I et al] I Lee, SV Date, AT Adai and EM Marcotte. *A probabilistic functional network of yeast genes*. Science **306** pp. 1555-8 (2004). PMID 15567862.

[Lee TI et al] TI Lee, NJ Rinaldi, F Robert, DT Odom, Z Bar-Joseph, GK Gerber, NM Hannett, CT Harbison, CM Thompson, I Simon, J Zeitlinger, EG Jennings, HL Murray, DB Gordon, B Ren, JJ Wyrick, JB Tagne, TL Volkert, E Fraenkel, DK Gifford and RA Young. *Transcriptional regulatory networks in Saccharomyces cerevisiae*. Science **298** pp. 799-804 (2002). PMID 12399584.

[Libby] P Libby. *Inflammation in atherosclerosis*. Nature **420** pp. 868-874 (2002). PMID 12490960.

[Luscombe et al] NM Luscombe, MM Babu, H Yu, M Snyder, SA Teichmann and M Gerstein. *Genomic analysis of regulatory network dynamics reveals large topological changes*. Nature **431** pp. 308-12 (2004). PMID 15372033.

[Lusis] AJ Lusis; *Atherosclerosis*. Nature **407** pp. 233-241 (2000). PMID 11001066.

[PubMed] PubMed: `http://www.ncbi.nlm.nih.gov/entrez/query.fcgi`
(Visited July 2004.)

[Seo et al] D Seo, T Wang, H Dressman, EE Herderick, ES Iversen, C Dong, K Vata, CA Milano, F Rigat, J Pittman, JR Nevins, M West and PJ Goldschmidt-Clermont. *Gene expression phenotypes of atherosclerosis*. Arterioscler Thromb Vasc Biol. **24** pp. 1922-1927 (2004). PMID 15297278.

[Shatkay and Feldman] H Shatkay and R Feldman. *Mining the biomedical literature in the genomic era: an overview*. J Comput Biol. **10** pp. 821-55 (2003). PMID 14980013.

[SGD] K Dolinski, R Balakrishnan, KR Christie, MC Costanzo, SS Dwight, SR Engel, DG Fisk, JE Hirschman, EL Hong, R Nash, R Oughtred, CL Theesfeld, G Binkley, C Lane, M Schroeder, A Sethuraman, S Dong, S Weng, S Miyasato, R Andrada, D Botstein and JM Cherry. *Saccharomyces Genome Database.* `http://www.yeastgenome.org/`
`ftp://ftp.yeastgenome.org/yeast/`
(Visited July 2004.)

A fool-proof method for sculpting an elephant: first, get a huge block of marble; then you chip away everything that doesn't look like an elephant.

Unknown

A The appendix

ALGORITHMS AND PSEUDO CODE is what this appendix is about. We displays some of the algorithms used in creating the random graphs. A short section with pseudo code illustrating the general structure of the program used to scan for cooccurances is included as well as the entity relations schema of the database.

A.1 Creating a random tree

Imagine we have a set of nodes N. With this set we wish to create the random tree $T = \{N, E\}$ where E is the set of edges for T. We can easily create the set of potential edges E_p by listing all possible pairs of nodes in N. The algorithm used to construct a quasi ER graph is the following:

Step 0: Initially we create sets of nodes, one set for each node in N. These sets, M_i, corresponds to with what other nodes the node i is connected (its neighbors and its neighbors neighbors and so on). Initially: $M_i = \{i\}, \forall i \in N$.

Step 1: Pick an edge $e_{i,j}$ at random from E_p and remove it from E_p.

Step 2: If M_i contains any element in M_j: discard $e_{i,j}$. Otherwise we would create a loop.

Step 3: If we keep $e_{i,j}$, add it to E', and for all elements p in $M_i \cup M_j$ let $M_p = M_i \cup M_j$. All nodes on each side of the new edge are now considered to be communicating.

Step 4: If $M_1 = N$ or $|E'| = n - 1$ or $|E_p| = 0$ break, else go to step 1.

A.2 Creating a random network

This algorithm creates quasi ER graphs that do differ from true ER graphs, but the difference is so small that it really does not matter.

If we want to add k edges to a random tree this is one possible algorithm (If k is negative we remove edges at random instead):

Step 0: Create a random tree as described above.

Step 1: Pick two nodes n_i and $n_j \neq n_i$ at random from N. Consider the edge $e = e_{min(i,j),max(j,i)}$.

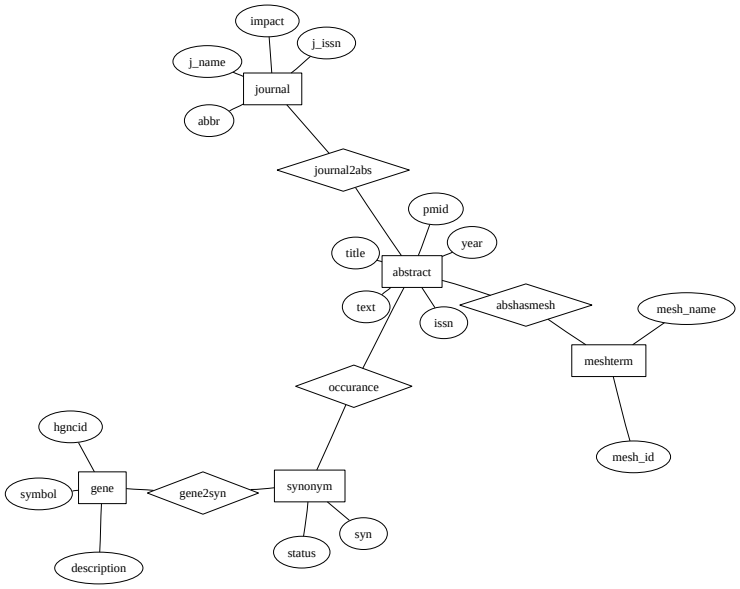

Figure A.1: A selected part of the entity-relation diagram of the database used to produce this thesis. The rectangles are entities (gene, synonym, abstract, journal and MeSH term) with its attributes (for example a gene has the attributes hgncid, symbol and description) in ellipses and the relations between entities are denoted by diamonds (a synonym may for example have an occurrence with an abstract).

Step 2: If we already have the edge e discard it (for negative values of k we could here remove it from the net instead).

Step 3: If we keep e, add it to E'. Set $k := k - 1$.

Step 4: If $k = 0$ or $|E'| = \frac{(n-1)n}{2}$ or $|E_p| = 0$ break, else go to step 1.

A.3 The database

The best way to describe a database is with its entity-relation diagram, this is displayed in figure A.1.

A.4 Pseudo code

Of great importance in this thesis is the database where for example all occurrences are stored. This little example of pseudo code will demonstrate the principles of going from a set of abstracts to the database.

```
for each abstract in abstract_set:
   for each meshterm in abstract:
      add (meshterm,pmid) to database.abshasmesh
   end for
   for each synonym in synonymlist:
      if synonym is included in body or title of abstract:
         translate synonym to gene
         add (gene,pmid) to database.occurance
```

```
      end if
   end for
end for
```

The implementation above is heavily suboptimal! For example: there are often less than 40000 words in an abstract so using all synonyms to scan is a bad idea. Instead one should use look for each word in the abstract in a hashtable of synonyms.

The other important part is of course going from a set of MeSH terms to a list of cooccurrences:

```
create empty hashtable pmids of integer

create empty hashtable cooccurrences of pairs of strings to integer

for each meshterm in meshterm_set:
   from database.abshasmesh(meshterm) get temp_pmids
   for each pmid in temp_pmids:
      if pmid not in pmids:
         to pmids add pmid
      end if
   end for
end for

for each pmid in pmids:
   from database.occurance get temp_genes
   if temp_genes has more than 2 items:
      for all pairs pair of items in temp_genes
         sort(pair)
         if cooccurrences has pair:
            cooccurrences(pair) += 1
         else:
            cooccurrences(pair) = 1
         end if
      end for
   end if
end for
```