In silico discoveries for biomedical sciences

Haagen, H. van

Chapter 2

In silico knowledge and content tracking

H.H.H.B.M. van Haagen, B. Mons

Chapter 9, Methods in Molecular Biology: In Silico Tools for Gene Discovery, Springer 2010

Abstract
We give a brief overview of a text-mining pipeline and the techniques allow explicit and implicit knowledge to be extracted from large text collections. First, a given ontology is used to tag terms in text as machine-readable concepts. Second, concepts are associated with each other using 2x2 contingency tables and test statistics. Third, from the contingency tables informative pair-wise links between concepts can be recovered. These links may be explicitly stated or implied through indirect associations. Fourth, validation techniques such as ROC curves and retrospective studies can be used to quantify the performance of the information extraction and knowledge discovery process. Lastly, we discuss methods combining text information with various non-textual data sources such as microarray expression data.
We conclude with a brief look at future directions for text-mining and knowledge discovery on the internet at large.

Keywords: text-mining, data-mining, information retrieval, disambiguation, retrospective analysis, ROC curve, prioritizer, ontology, semantic web

1 Introduction

The amount of biomedical literature is growing tremendously. It has become impossible for researchers to read all publications in their moving field of interest, which forces them to make a stringent selection of relevant articles to read. For the actual knowledge discovery process, which is in essence a systematic association process over an expanding number of interrelated concepts, life scientists increasingly rely on the computer. This stringent reduction of the percentage of relevant articles that can actually be 'read' has the disadvantage that relevant information from non-selected articles can be missed. The largest database of recorded biomedical literature is PubMed, which contains over 14 million articles published in the last 30 years (from 1980 till 2010). Besides the literature there are many other resources ranging from curated databases to online blogs, digital books recorded in libraries, and any text information that can be found via a search engine like Google.

The field that deals with automated information extraction from text is called text-mining. Text-mining on its own is a challenging field of research that intensively has been further developed over the last years. Computer systems have been developed based on natural language processing; a method of processing any sentence into its building blocks such as the subject, verbs, and nouns. Other methods are based on word tagging. PubMed for instance uses the words in a

search query and matches it with words found in abstracts with no additional information how the words are related with other words in text. In this chapter we describe the concept based method for automated information extraction from text.

2 Concept based text-mining

For concept based text-mining three 'ingredients 'are needed: (1) text data (2) a word tagger, and (3) a terminology system, mostly controlled vocabulary, or ontologies.
For biomedical text data, normally the abstracts recorded in PubMed are chosen. Reasons for this are that this is the greatest source of recorded literature, the abstracts are publically available and free to download, and the information density of abstracts is higher than that of full text documents (*1*).
Words in text are recognized by a so called word tagger and mapped to a concept identifier (*2*). In order to do so we first need to understand what a concept is. A concept is a unit of thought meaning that people agree that they share information about one and the same thing. A concept has terms and other 'tokens' that 'refer' to it. It can have synonyms, abbreviations, but also for instance Uniform Resource Identifiers (URI's) or accession numbers.
For instance, there exists a protein called dystrophin. When the gene encoding for this protein is mutated it can cause diseases such a Duchenne muscular dystrophy or Becker muscular dystrophy. Dystrophin normally is abbreviated to DMD. DMD (either in italic) also refers to the gene or the disease. Dystrophin is stored in databases like Entrez Gene (http://www.ncbi.nlm.nih.gov/gene) with the accession number 1756 and Uniprot Knowledge Database (http://www.uniprot.org/) with accession number P11532. The words dystrophin, DMD, 1756, and P11532 all refer to one and the same concept (we treat a gene and a protein as the same concept). The tagger maps the words to the concept identifier for dystrophin.
Lastly the synonyms, abbreviations, accession numbers, and concept identifiers are stored in an Ontology. The most common vocabulary for the biomedical field is the unified medical language system (UMLS)(*3*). An Ontology may be field specific. If only drug information from text needs to be extracted a drug-vocabulary is used instead of the whole vocabulary with all medical concepts.

3 Classical direct relationship detection

Once a text-mining system has been developed and concepts in text are recognized and stored in a database the question becomes what to do with this tagged text data? The main question is which two concepts are significantly related. The relationship between two concepts can be of any kind. In biology these are the most common ones we chose as examples: (1) two proteins that have a molecular

interaction, (2) a mutated gene that causes a disease, (3) a protein that has a particular function and (4) a drug that treats a disease or has a (adverse) side effect. Any relationship between two concepts can be seen as a triplet with a subject, predicate and object. An example of a triplet is protein dystrophin (subject) interacts with (predicate) protein ankyrin 2 (object).
The statistical way to define the strength of relationship between two concepts is by making a 2x2 contingency table (or frequency table). The table below gives an example for concepts X and Y.

	X	Not X
Y	A	B
Not Y	C	D

A are the number of documents where both concept X and Y are co-mentioned. *B* are the number of documents where concept Y occurs but not concept X. *C* is the reverse version of *B* and *D* are the number of documents where X and Y are not mentioned. Any statistical test can be applied to this table such as the likelihood ratio test, chi-squared test or the uncertainty coefficient. If X and Y are frequently co-mentioned together (*e.g. A* is a relatively large number) and the concepts are not exceptionally generic so that they occur frequently in text (*e.g B* and *C* are small) then the two concepts may be significantly related. There are many text-mining systems available based on direct relationship detection such as IHOP**(*4*)**, PubGene**(*5*)**, and systems where text-mining is an integral part such as STRING**(*6*)**, FunCoup**(*7*)**, and Endeavour**(*8*)**.

4 Implicit information extraction via concept profiling

The classical direct relationship detection method has the disadvantage that concepts that are not co-mentioned together are missed, while they still might be related to each other. This could be due to the reason that related concepts are stored in full text (frequently not freely available for mining) and not in the abstract or that concepts are related but no-one made the link yet. Via indirect links between terms in text, terms can still be related to each other even when they have never been co-mentioned - **(*9*)**. This we call implicit information extraction. Swanson et.al. **(*10*)** were the first to demonstrate that this approach works by linking the treatment of Raynaud's disease with fish oil. Van Haagen et. al. **(*11*)** demonstrated this idea further by predicting protein-protein interactions. They predicted the physical interaction between calpain 3, which causes a form of muscular dystrophy, and parvalbumin B, which is found mainly in skeletal muscle. Those two proteins were strongly linked via the intermediate concept dysferlin, which is a protein.

Concept profiling contains the following steps (see Fig. 1). First for a concept X (*e.g* a gene, a chemical or drug) the documents are selected wherein X appears. Next all other concepts that are co-mentioned with X are processed using the direct relationship detection method described previously (Fig. 1b). The 2x2 table information for each concept pair is stored in a profile. This concept profile for X is basically a vector of N dimensions. N are the number of concepts that are co-mentioned with X. Each entry in the vector is a number associating concept X with another concept (taken from a 2x2 table, Fig. 1a). Computation of the 'conceptual association' between two concepts can now be performed by matching their respective concept profiles by vector matching (Fig. 1c). Any distance measure can be used for this matching**(*9*)** such as the inner product, cosine, angle, Euclidean distance or Pearson's correlation. If two concept profiles have many concepts in their individual concept profiles in common, *e.g.* many implicit links, then the two concepts may be related to each other. A webtool is available, dubbed 'Anni', for implicit information extraction by concept profiling**(*12*)**. In the next section we will describe how to validate text-mining approaches and the amount of relatedness.

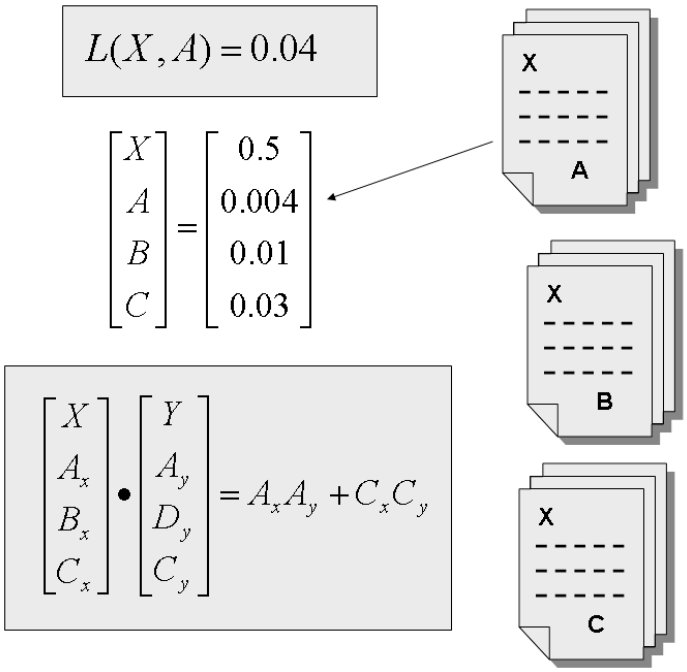

Figure 1. Basic scheme for concept based profiling. (a) Example of a likelihood function calculated between concept X and A. Information is taken from a 2x2 contingency table. The score reflects the strength of association between X and A. (b) Documents selected where concept X appears and is co-mentioned with other concepts. For a concept the documents are selected and transformed into a test statistic using a 2x2 contingency table. (c) The inner product score between two

concept profiles. The score is only calculated over the concepts the two profiles have in common.

5 Cross validation within text-mining and other performance measures

5.1 Defining a positive and a negative set.

In the previous sections we described how to extract relationships (content) between concepts from text either with direct relationship detection or concept profiling. Once a system is designed it needs to be tested to evaluate its performance in extracting or predicting relationships. To enable this step we need data to train the system and after training testing it. For instance data on protein function can be collected from the Gene Ontology **(*13*)** and data on gene-disease relationships from OMIM (http://www.ncbi.nlm.nih.gov/omim). Here we describe an example of the relationship type protein-protein interactions (PPIs). PPIs can be collected from online databases such as UniProt**(*14*)**, DIP**(*15*)**, BioGrid**(*16*)** and Reactome**(*17*)**. These samples of curated protein-protein interactions are labeled as positives instances. These positives instances are compared with negative instances to see if the text-mining system can discriminate between the two groups. In biology research no databases exists that stores samples of negatives instances, *e.g.* two proteins that have been confirmed not to interact. Normally generating negative instances is done by selecting random pairs from a group of proteins**(*18*)**.

5.2 Receiver operating characteristics curves

Receiver operating characteristics (ROC) curves are often used to evaluate the performance of a prediction algorithm **(*19*)**. A ROC curve is a graphical plot of the true positive rate (sensitivity) on the y-axis versus the false positive rate (1 − specificity) on the x-axis (see Fig. 2b). The ROC curve is defined for a binary classifier system (the positive and negative set described in section 5.1) as its discrimination threshold is varied. This measure is often used in information retrieval and it can be explained as a system design that collects as much information as possible (in terms of true positives) while at the same time reducing the noise (the false positives). A ROC curve is constructed as follows; in Fig. 2a the distributions of positive and negative instances are given and in Fig. 2b its corresponding ROC curve. The threshold that discriminates between the two groups is varied from the highest match score (x-axis Fig. 2a) value to the lowest. Each threshold corresponds to a true positive and false positive rate in ROC space. In Fig. 2a all the way up to the right on the x-axis is the threshold (around 7) where no true or negative instances pass this threshold. Therefore the true positive and false positive rates are both zero, resulting in the point (0,0) in ROC space (Fig. 2b bottom left corner). Then the threshold as a slider is moved to the right. At each

point a number of positive and negative instances will pass the threshold resulting in a point in ROC space anywhere between 0 and 1 on both axes. Finally the threshold reaches the extreme left point on the x-axis (around -2, Fig. 2a). Here all positive and negative instances pass this threshold. This corresponds with the point (1,1) in ROC space (top right corner Fig. 2b).
To translate the ROC space to a single measurement for performance we calculate the Area under the ROC curve (AuC). The AuC value normally varies between 0.5 and 1.
If a system shows a random behavior (*e.g.* two completely overlapping distributions, no discrimination between positive and negative set) the ROC space results in a straight line from the point (0,0) to (1,1). This corresponds with an AuC of 0.5. If a system behaves like a perfect classifier the ROC curve starts at point (0,0) and moves up to point (0,1) (*e.g.* first all positive instances are predicted) then it moves from point (0,1) to point (1,1) (*e.g.* all negative instances are predicted). This corresponds with an AuC of 1. The AuC for the example in Fig. 2 is 0.92.

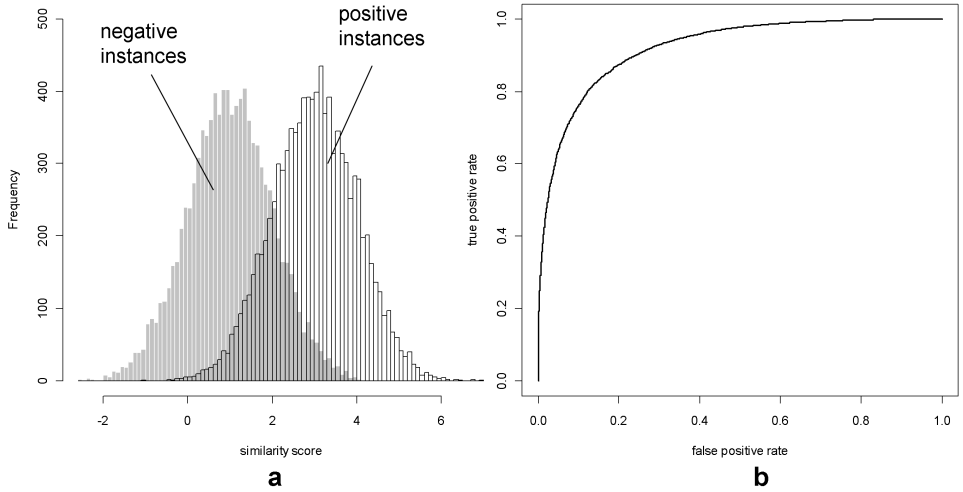

Figure 2. Histogram and its corresponding ROC plot. (a) the distribution of the positive and negative set. (b) a ROC curve with an AuC of 0.92.

5.3 Cross validation and bias

The performance of an associative *in silico* discovery system is tested using cross-validation(***20***). A system is first trained using training data. Then it is tested using test data. There is no explicit data for testing only, nor is their data used only for training. There is just data. Therefore a part of the data is selected for training and the remaining part for testing. The way to select the training and test data is arbitrary. Here we describe the most common approach of cross validation the 10-

fold CV. The first step (1) is to randomly shuffle the samples in your dataset (both positive and negative instances). (2) Then the dataset is divided into 10 equally sized subsets. Each piece contains samples of the positive and negative set. (3) In one iteration, 9 of the 10 subsets is used for training and the remaining subset is used for testing. (4) Step three is repeated until each subset is used once for testing.
An extremely important step during cross validation is to make sure that none of the test data is used during training. Else this would introduce a bias and gives an overestimation of the true performance. Within the field of text-mining and biology this seems virtually impossible. Most of the data stored in curated databases, such as protein-protein interactions or gene-disease relationships recorded in OMIM are based on published articles. This means that positives instances in the test set are based on articles that are also used to train a text-mining system. Other data sources also have this problem. For instance the Gene Ontology contains functional descriptions for a protein that are normally also based on literature evidence. In order to evaluate prediction performance it is therefore more appropriate to make use of a retrospective analysis

5.4 Retrospective validation

Before we explain the basics of retrospective validation, we need to distinguish between two types of prediction. The first one is prediction of current knowledge stored in databases. This knowledge is already known and the system recovers what is stored in these databases. For this, the cross-validation approach described above is useful.
The second one is the prediction of new and as yet unforeseen knowledge. This means 'implicit' knowledge that is not recorded in any database that cannot be explicitly found in text. To simulate the prediction of these 'hidden associations' a retrospective validation is done. First a time interval is defined when data is stored in a database. For PubMed this could for instance be all the abstracts of articles published between 1980 and January 2010. The second step would be to select test data published after a certain date, for instance all protein-protein interactions recorded in databases from January 2007 until January 2010. The third step is to train the text-mining system before that date using all data before January 2007. The last step is to evaluate what test samples were predicted before January 2007 that became only explicit (also in the literature) knowledge after January 2007. In other words, protein-protein interactions that could be found by simple co-occurrence before the 'closure date', but were not added to the databases yet, should not be counted as true predictions. In this evaluation there is no procedure to repeat these steps multiple times like with cross-validation. This means that no standard error on the performance can be calculated.

5.5 Prioritizers

Another way to view a ROC curve is as a prioritized list. The ROC curve is constructed by varying the threshold. The samples (*e.g.* protein pairs either a PPI or random) are ranked from the highest match score to the lowest. Going down in this ranked list from the top prediction to the lowest is done by walking over the ROC curve from point (0,0) to point (1,1). Experimental biologists are mainly interested in what is predicted in the top, *e.g.* the most likely predictions. Prioritizers are useful to evaluate where your test samples rank in the top. A ROC curve can also be plotted on the absolute scales of true positives and false positives by translating a prioritized list in a graphical way. Figure 3 shows an example of 20 ranked samples and its corresponding ROC curve. This curve is also called a ROC10 curve. It reflects the amount of true positive predictions (baits) at a fixed number of false positives (the costs), in this case 10. You can vary this threshold and define for instance a ROC50 or ROC100 curve.

sample	match score	label
s1	4.960111897	0
s2	4.662243252	0
s3	4.661449007	1
s4	4.589346313	0
s5	4.581602664	0
s6	4.438759168	0
s7	4.379399852	0
s8	4.377182023	0
s9	4.320084484	1
s10	4.303145807	0
s11	4.299505092	1
s12	4.259483679	0
s13	4.249962717	1
s14	4.230262428	1
s15	4.188873972	1
s16	4.179967181	1
s17	4.177931731	1
s18	4.176843766	0
s19	4.163993686	1
s20	4.118021526	1

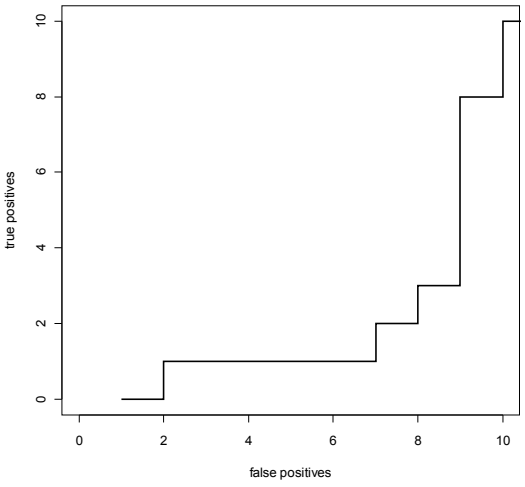

Figure 3. Prioritized list of 20 samples and its corresponding ROC10 curve.

6 Extending text-mining systems with other databases: data-mining

Text-mining actually is a subdivision of the broader field of data-mining. Data-mining is the field of research to extract any kind of information from a variety of resources. For instance, there are many data sources available for proteins. Besides the literature, there exists information in curated databases, microarray expression data(***21, 22***), domain interaction databases(***23***), functional annotations from the Gene Ontology, phylogenetic trees and sequence data. There are many tools and techniques available for data-mining on databases but they all share a common idea. To combine all information from several distinct data sources into one should reveal more information than can be recovered by the mining of each data source alone. Data-mining basically is a two step approach. The first step is to define a match or evidence score for every data source that is included in the system. For instances a microarray dataset may be transformed into a data matrix by calculating Pearson correlations between any two expression profiles for proteins or genes. The second step is to combine each evidence score for a data source into a single score. This can be done, for instance, using a Bayesian classifier. For protein-protein interactions there are several resources available based on data-mining techniques such as STRING(***6***), FunCoup(***7***), IntNetDB(***24***), and Prioritizer(***25***).

7 Beyond data-mining and scalable technology for the internet: the semantic web

Data-mining and text-mining are fields of technology that are used for the future web 3.0 technology: the semantic web (SW). The first trend in web technology (or web 1.0) included the static webpages that made the first version of the internet. No information exchange was possible, just readable plain text pages. The second trend (web 2.0) made it possible for users to interact with the internet. Think of uploading movies to YouTube, or writing your blog online and online shopping with a credit card. Web 2.0 is really the most unstructured and scattered form of information. Therefore, the new trend became web 3.0. It will structure the internet into a network of concepts and relationships between these concepts. Other terms for web 3.0 are the concept web or the semantic web. One of the goals of the web is to present information in a computer readable compact format instead of the current webpages that are retrieved after a search query. The predictions that are made using concept profiles or other technologies will be part of this SW.

The best known data model for the SW is RDF (resource description framework). RDF is used to translate any kind of date into a triple format. The ontologies used in webtechnology are mainly built using OWL (Web Ontology Language). The semantic web project is extremely large and it is very difficult to keep it scalable. There is now an ongoing project called the Large Knowledge Collider (LarKC). It builds the semantic web with all the current state of the art technology that is out

there (machine learning, information theory, pattern recognition, first order logic). All information on LarKC can be found on http://www.larkc.eu.

8 References

1. Schuemie, M. J., Weeber, M., Schijvenaars, B. J., van Mulligen, E. M., van der Eijk, C. C., Jelier, R., Mons, B., and Kors, J. A. (2004) Distribution of information in biomedical abstracts and full-text publications, *Bioinformatics 20*, 2597-2604.
2. Schuemie, M. J., Jelier, R., and Kors, J. A. (2007) Peregrine: Lightweight gene name normalization by dictionary lookup, in *Biocrative 2 workshop*, pp 131-140, Madrid.
3. Bodenreider, O. (2004) The Unified Medical Language System (UMLS): integrating biomedical terminology, *Nucleic Acids Res 32*, D267-270.
4. Hoffmann, R., and Valencia, A. (2004) A Gene Network for Navigating the Literature, *Nature Genetics 36*, 664.
5. Jenssen, T. K., Laegreid, A., Komorowski, J., and Hovig, E. (2001) A literature network of human genes for high-throughput analysis of gene expression, *Nat Genet 28*, 21-28.
6. Jensen, L. J., Kuhn, M., Stark, M., Chaffron, S., Creevey, C., Muller, J., Doerks, T., Julien, P., Roth, A., Simonovic, M., Bork, P., and von Mering, C. (2009) STRING 8--a global view on proteins and their functional interactions in 630 organisms, *Nucleic Acids Res 37*, D412-416.
7. Alexeyenko, A., and Sonnhammer, E. L. (2009) Global networks of functional coupling in eukaryotes from comprehensive data integration, *Genome Res 19*, 1107-1116.
8. Aerts, S., Lambrechts, D., Maity, S., Van Loo, P., Coessens, B., De Smet, F., Tranchevent, L. C., De Moor, B., Marynen, P., Hassan, B., Carmeliet, P., and Moreau, Y. (2006) Gene prioritization through genomic data fusion, *Nat Biotechnol 24*, 537-544.
9. Jelier, R., Schuemie, M. J., Roes, P. J., van Mulligen, E. M., and Kors, J. A. (2008) Literature-based concept profiles for gene annotation: the issue of weighting, *Int J Med Inform 77*, 354-362.
10. Swanson, D. R. (1986) Fish oil, Raynaud's syndrome, and undiscovered public knowledge, *Perspect Biol Med 30*, 7-18.
11. van Haagen, H. H. H. B. M., t Hoen, P. A. C., Botelho Bovo, A., de MorrÃ©e, A., van Mulligen, E. M., Chichester, C., Kors, J. A., den Dunnen, J. T., van Ommen, G.-J. B., van der Maarel, S. r. M., Kern, V. c. M., Mons, B., and Schuemie, M. J. (2009) Novel Protein-Protein Interactions Inferred from Literature Context, *PLoS ONE 4*, e7894.

12. Jelier, R., Schuemie, M. J., Veldhoven, A., Dorssers, L. C., Jenster, G., and Kors, J. A. (2008) Anni 2.0: a multipurpose text-mining tool for the life sciences, *Genome Biol 9*, R96.
13. Gene Ontology, C. (2000) Gene ontology: tool for the unification of biology, pp 25 - 29.
14. (2009) The Universal Protein Resource (UniProt) 2009, *Nucleic Acids Res 37*, D169-174.
15. Salwinski, L., Miller, C. S., Smith, A. J., Pettit, F. K., Bowie, J. U., and Eisenberg, D. (2004) The Database of Interacting Proteins: 2004 update, *Nucleic Acids Res 32*, D449-451.
16. Stark, C., Breitkreutz, B. J., Reguly, T., Boucher, L., Breitkreutz, A., and Tyers, M. (2006) BioGRID: a general repository for interaction datasets, *Nucleic Acids Res 34*, D535-539.
17. Matthews, L., Gopinath, G., Gillespie, M., Caudy, M., Croft, D., de Bono, B., Garapati, P., Hemish, J., Hermjakob, H., Jassal, B., Kanapin, A., Lewis, S., Mahajan, S., May, B., Schmidt, E., Vastrik, I., Wu, G., Birney, E., Stein, L., and D'Eustachio, P. (2009) Reactome knowledgebase of human biological pathways and processes, *Nucleic Acids Res 37*, D619-622.
18. Ben-Hur, A., and Noble, W. (2006) Choosing negative examples for the prediction of protein-protein interactions, p S2, BMC Bioinformatics.
19. Fawcett, T. (2003) ROC Graphs: Notes and Practical Considerations for Data Mining Researchers, *Hewlett-Packard Company*.
20. Wessels, L. F., Reinders, M. J., Hart, A. A., Veenman, C. J., Dai, H., He, Y. D., and van't Veer, L. J. (2005) A protocol for building and evaluating predictors of disease state based on microarray data, *Bioinformatics 21*, 3755-3762.
21. Obayashi, T., Hayashi, S., Shibaoka, M., Saeki, M., Ohta, H., and Kinoshita, K. (2008) COXPRESdb: a database of coexpressed gene networks in mammals, *Nucleic Acids Res 36*, D77-82.
22. Su, A. I., Wiltshire, T., Batalov, S., Lapp, H., Ching, K. A., Block, D., Zhang, J., Soden, R., Hayakawa, M., Kreiman, G., Cooke, M. P., Walker, J. R., and Hogenesch, J. B. (2004) A gene atlas of the mouse and human protein-encoding transcriptomes, *Proc Natl Acad Sci U S A 101*, 6062-6067.
23. Mulder, N. J., Apweiler, R., Attwood, T. K., Bairoch, A., Bateman, A., Binns, D., Biswas, M., Bradley, P., Bork, P., Bucher, P., Copley, R., Courcelle, E., Durbin, R., Falquet, L., Fleischmann, W., Gouzy, J., Griffith-Jones, S., Haft, D., Hermjakob, H., Hulo, N., Kahn, D., Kanapin, A., Krestyaninova, M., Lopez, R., Letunic, I., Orchard, S., Pagni, M., Peyruc, D., Ponting, C. P., Servant, F., and Sigrist, C. J. (2002) InterPro: an

integrated documentation resource for protein families, domains and functional sites, *Brief Bioinform 3*, 225-235.

24. Xia, K., Dong, D., and Han, J. D. (2006) IntNetDB v1.0: an integrated protein-protein interaction network database generated by a probabilistic model, *BMC Bioinformatics 7*, 508.
25. Lage, K., Karlberg, E. O., Storling, Z. M., Olason, P. I., Pedersen, A. G., Rigina, O., Hinsby, A. M., Tumer, Z., Pociot, F., Tommerup, N., Moreau, Y., and Brunak, S. (2007) A human phenome-interactome network of protein complexes implicated in genetic disorders, *Nat Biotechnol 25*, 309-316.