Biological Database Normalization by Sequence Alignment

Aaron Elkiss

ABSTRACT

The Michigan Molecular Interactions (MiMI) database contains protein interaction data from many distinct sources. Frequently, the same protein is referred to in different external databases by different identifiers, so it is difficult to determine when different records refer to the same object. The normalization problem is addressed by creating an extended MiMI database that integrates MiMI with the large publicly available protein sequence databases and using BLAST to enable the user to find interactions for proteins with high similarity to any given query protein. I discuss the challenges of rigorously evaluating this approach but conclude that it should prove to be a useful complement to MiMI. A prototype of the system is available for use at `http://oriole.eecs.umich.edu/mimi`.

1. INTRODUCTION

The problem of integrative bioinformatics, that is, combining biological data from disparate sources, has become increasingly important as the number of data sources grows. Researchers should ideally be able to search all available information of a given type through a single, consistent interface. The MiMI project [10] tackles this problem in the area of protein interactions. One of the major challenges for the integration of disparate data sources is normalization; that is, determining whether two records refer to the same underlying entity. In the biological domain, the underlying entity is determined by a nucleotide or amino acid sequence. Thus, considering sequence similarity should provide us with a way to find determine whether two database records cite to the same biological entity.

In this paper I give a brief review of the relevant biology, present a database of biological interactions which several other data sources and discuss some issues with it, and then discuss methods for normalizing this database. I consider several possibilities for evaluating the performance of the normalization scheme and conclude by suggesting other domains where similar normalization techniques could be interesting.

1.1 Biological Preliminaries

In order to understand the data to be normalized, it is necessary to review the relevant biology (see [4] for a comprehensive introduction to cell and molecular biology). The most important fact is that the vast majority of interacting molecules in the databases of interest are not atomic entities. Just as novels are composed of sequences of letters from a fixed alphabet, as computer programs are composed of a sequence of small instructions, as musical pieces are made of a sequence of notes, so proteins are polypeptides, composed of a sequence of amino acids. Thus, proteins have a structure that can be analyzed above the level of individual atoms but below the level of entire molecules. The sequence of amino acids determines the chemical properties and three-dimensional structure of the protein, which in turn determines (in ways we do not yet fully understand) what other molecules the protein interacts with.

Proteins are built in individual cells by ribosomes. However, ribosomes must be told what protein to build. DNA and RNA perform this function. Like proteins, DNA is a polymer. It consists of a long sequence of four different possible nucleotides. Each sequence of three nucleotides (a *codon*) codes for a particular amino acid. There are 64 (4^3) possible codes, but only 20 different amino acids. Thus, most amino acids are coded for by more than one codon, making some mutations at the DNA level *silent*.

DNA can be thought of as analogous to the binary code for an entire executable program. The executable program may contain additional data necessary for execution but not actually part of the program, for example strings to print or images to display. Likewise, DNA contains a great deal of material that does not seem to code for proteins and whose function is not yet well-understood. In contrast, proteins can be thought of as analogous to the assembly code listing for a program - still very low-level, but easier to analyze than a string of bytes.

Even at the protein level, some mutations are more important than others. Some amino acid substitutions may have greater effect than others. For example, substituting one hydrophilic amino acid for another will have less effect than substituting a hydrophilic amino acid for a hydrophobic one. In the first case both proteins will still be attracted to water and thus will likely end up close to the outside of the final structure of the protein. In the second case, the mutated protein no longer is attracted to water and will attempt to be on the inside of the final structure, so the final structure might look quite different. This may significantly affect what other molecules the protein interacts with.

Each substitution from some amino acid to another amino acid can be given a score based on the frequency they are substituted in sets of proteins known to be related to some extent, generally the same protein in different species. Proteins more likely to be substituted in this way than one would expect by chance (i.e. assuming substitutions occur uniformly) are called *conserved*. Those less likely to be substituted than one would expect by chance are called *non-*

conserved.

Two sequences are called *homologous* if they are derived from a common ancestor. For example, many proteins in mouse and human are homologous; they derive from some common mammalian ancestor. Homologous protein sequences will generally be conserved for most of their length; this is why sequence similarity is a good predictor for homology.

1.2 The MiMI Database

MiMI (*Michigan Molecular Interactions*) [10] is a database consisting of molecules and the interactions between them. The vast majority of molecules are proteins. These are the only molecules considered in this study. MiMI integrates interaction data from a variety of publicly available data sources, including DIP [17], BIND [12], GRID [8], HPRD [13], and others. The version of the MiMI database used comprises 66,713 molecule records and 138,454 interaction records. Each molecule in MiMI has at least one external identifier, which can be used to reference the molecule in other data sources.

So far as possible, interactions from different data sources are combined into a single MiMI entry based on the external database ID. Other than HPRD, the external IDs are not primary keys in the data sources MiMI integrates. They are instead references into databases whose primary function is to hold sequence data rather than interaction data. The most common identifiers are GIs assigned by NCBI [1] and NCBI RefSeq identifiers [16]. It is frequently the case that different interaction databases may refer to the same protein by non-overlapping sets of external IDs. For example, HPRD asserts that a particular protein (human androgen receptor) has RefSeq ID `NP_000035`, among other external identifiers, while BIND asserts that a particular protein has LocusLink ID `367`. None of the external IDs provided by BIND or HPRD for this protein overlap, but LocusLink ID `367` also maps to `NP_000035`. Thus both the BIND and HPRD data should be merged to the same entry in MiMI, but are not. Thus, a first step at normalization might be to collect lists of equivalent external IDs and then compute the connected components of the graph induced by this equivalence relation.

However, this does not entirely solve the problem. What should count as the same protein? Only proteins with exactly the same sequence? Only the same protein from a particular organism, but allowing for normal genetic variation? Homologous proteins from some group of organisms? The external ID mapping typically operates at the level of protein within a particular organism, but it can be the case that the same sequence is assigned multiple IDs even within the same database. Continuing with the androgen receptor example, the HPRD entry has a reference to GI `105325` and the BIND entry has a reference to GI `178628`. These GIs have the exact same protein sequence, but for whatever reason, `105325` has been discontinued and been replaced – although by `113830`, not `178628`!

Thus, to a large extent the problem of duplicate entries in MiMI is a reflection of problems with the underlying data sources rather than flaws with the design of MiMI itself. However, one would like to compensate for these problems in as robust a way as possible. The solution is to allow searching MiMI based on sequence similarity. MiMI itself does not contain sequence data for each protein, so the external ID mapping is used to get the sequence data.

It should be noted that simply finding similar proteins is not an ideal solution. The true task is to find all the interactions in the database a given protein participates in, or more precisely to determine given two proteins A and B and the list of interactions of B, determine which of the interactions A also participates in. This is a somewhat more constrained problem than the general problem of determining if two given proteins interact, but still well outside the scope of this study. There is a great deal of ongoing work in the area of protein interaction prediction (see [19] for a recent survey.) The extended MiMI database simply returns the list of similar proteins and their interactions and leave it to the user to interpret the data. Section 4.1 mentions some possible refinements to this strategy.

1.3 BLAST

Given the set of sequences of proteins in the MiMI database and some sequence to query with, one needs some similarity metric. Ideally, this metric should be efficient to compute and should correlate well with homology. The BLAST (Basic Local Alignment Sequence Tool) algorithm [5] serves this purpose.

By default BLAST reports all alignments that are unlikely to have occurred by chance. However, we are interested in a much stronger notion. If we accept the assumption that homologous proteins are likely to participate in the same interactions, we would like all the proteins returned for a search to be homologous.

The score that BLAST reports for an alignment is the simply the sum of the score of each individual aligned residue pair. Identical and conserved pairs will contribute positively to the score; non-conserved pairs and gaps will contribute negatively. Thus the BLAST score will correlate strongly with the percentage of the query aligned, the percent identity, and the percent conserved. Additionally, the number of alignments E expected by chance with a given score for a given query and database is a simple function $E = kmne^{-\lambda S}$ of the score S, the size of the query and the database m and n, and constants of proportionality λ and k [6].

1.4 Related Work

The ATLAS database [18] is an effort similar to MiMI that integrates various biological databases. It integrates several sequence and function databases in addition to the interaction databases, but its user interface is significantly less user-friendly and informative than MiMI's, and it suffers from the same problems as MiMI with respect to interaction data (described in section 1.2). There is no integration with BLAST or any BLAST-like tool.

NCBI's HomoloGene [3] is a database of automatically detected homologous genes among 18 species. This data only covers a relatively small set of genes and is generally based on gene and not protein sequences, so although it could help in normalizing the MiMI data it is not a complete solution. It also suffers from the same weaknesses as other static clusterings.

NCBI's BLink [2] displays precomputed BLAST alignments for all proteins vs. all other proteins in NCBI's protein database. The functionality in this study was inspired by the idea of connecting a BLink-like interface with MiMI.

Ulysses [15], a very recent follow-on project to Atlas, uses a technique called interolog analysis to project interactions across species. Although the main focus is predicting in-

External DB	*Count*	*Source*
NR	3031987	`ftp://ftp.ncbi.nlm.nih.gov/blast/db`
RefSeq	1840189	`ftp://ftp.ncbi.nlm.nih.gov/blast/db`
SwissProt	183162	`ftp://ftp.ncbi.nlm.nih.gov/blast/db`
HPRD	22158	`http://www.hprd.org`
ORF	6704	`http://www.yeastgenome.org`

Figure 1: Count of sequences obtained from external databases. These sequences cover 94.8% of the records in MiMI.

External DB	*Count*
GI	7843231
RefSeq	4002974
LocusLink	3109018
GenBank	2352480
EMBL	580702
SwissProt	435796
DDBJ	389263
HPRD	104323
PDB	74338
UniGene	36592
PubMed	11745
OMIM	9173
ORF	6704

Figure 2: Count of external IDs obtained from external databases. These external IDs cover 94.8% of the records in MiMI.

teractions for human proteins based on homologous proteins in other organisms, this technique could be relevant to database normalization. Ulysses uses the HomoloGene information to group homologous genes. Similar to this study, evaluation is a challenge. User interface is more of a focus than in this study.

2. METHODS

2.1 Initial Database Integration

Sequence data and external ID mappings from several publicly available sequence databases were obtained as well as one database of interactions which also contained sequence data. These data sources are summarized in Figure 1 and 2. These sequences and external IDs covered 63,280 out of 66,713 molecules in MiMI. The remaining 3433 records were primarily (2,220 records) molecules from HPRD that had no associated external IDs or sequence data. Some of these molecules were not in fact even proteins – for example there are molecules which report HPRD IDs of "Sodium", "RNA" and even, mysteriously, "Mouse". There are also 1,155 LocusLink (EntrezGene) identifiers for which it was not possible to find associated proteins. Figure 3 lists the unmappable external IDs by external database. Figure 4 lists the count of external IDs in the MiMI database, again grouped by external database.

The existing MiMI database augmented with this additional sequence and identifier data is referred to as the *extended MiMI database*.

2.2 BLAST Parameters

External DB	*Count*
HPRD	2220
LocusLink	1155
GI	47
SGD	18
FlyBase	14
RefSeq	12
ORF	11
DIP	9
IPI	7
SwissProt	5
PIR	1

Figure 3: Count of unmappable external IDs in MiMI by external database. Adds up to 3499, slightly more than the 3433 unmappable MiMI records, since some of the unmappable records have more than one unmappable external identifier.

External DB	*Count*
RefSeq	40067
GI	30836
LocusLink	21668
SwissProt	18509
HPRD	18303
IPI	14555
UniGene	12016
OMIM	9176
DIP	7998
SGD	5417
ORF	4948
PIR	4540
FlyBase	4272
YPD	4105
WormBase	138

Figure 4: Count of all external IDs in MiMI by external database. Most MiMI records have more than one external identifier.

BLAST has many tunable parameters which affect the ultimate score for each alignment. [6] The extended MiMI database runs BLAST with the following parameters:

- `-e 1e-15` - The expected number of times to see an alignment with this score in the database by chance, i.e. assuming amino acids are chosen randomly, should be very close to zero.
- `-G 25 -E 2` - Gap opening and extension costs are chosen to be as expensive as possible. This is the most expensive recommended setting for protein-protein alignments. [6]
- `-A 10` - The BLAST algorithm works by aligning small initial windows and then expanding them. The current version of BLAST can find two separate windows in each possible hit and expand them simultaneously. Setting the `-A` parameter lower than the default of 40 forces the two windows to be closer together. For homologous sequences, one expects the sequences to be aligned more or less contiguously over most of their lengths. Thus, there is no reason to allow initial windows to be far apart.
- `-M BLOSUM80` - The `-M` parameter controls the score assigned to each aligned residue pair. The BLOSUM80 matrix was created by aligning protein blocks that have at least 80% similarity to each other. [14] This matrix is most stringent matrix provided in the BLAST distribution.

2.3 Searching the Database

The extended MiMI database can be searched in three ways: by external ID, by MiMI ID, and by sequence. It gathers the set of distinct protein sequences for the external ID or for all external IDs mapping to the MiMI ID, or the single provided sequence, and uses the above BLAST parameters to search for similar sequences. The returned sequences are grouped by the MiMI ID or IDs they refer to, and (by default) sort by BLAST score. If there is more than one matching sequence for a given MiMI ID, the highest score of any of the matches is used. The user can also sort by other criteria: percent of query aligned, E-value, percent identity, percent conserved. The iterative nature of interactive search means that the user can choose to refine the search with more appropriate criteria, although of course the hits should be returned in the most useful order by default.

For each result (MiMI molecule) the following is displayed: a graphical overview of each alignment with a tooltip with the external IDs for the hit sequence, a list of all external IDs for the MiMI molecule, a list of names for the MiMI molecule, and a list of the interactions the MiMI molecule participates in. The list of external IDs and names consists not only of those in the original MiMI database but also those external IDs and names found through the connected components described in section 1.2. Each interacting molecule has a link so that the extended MiMI database can immediately be searched for the interacting molecule, and each external ID has a link to search the external database the ID came from. See `http://oriole.eecs.umich.edu/mimi` for an example search; the results are too long to include as a figure.

The original version of MiMI contains additional information such as provenance and confidence scores for each interaction [10]. This additional information is not replicated in the results page, since ideally there would be a link to the original MiMI database for this information. However, one weakness of MiMI is that its internal identifiers do not remain consistent between releases of the database. The version provided for use in this studyis not the same version currently available on the MiMI web site; thus, it was not possible to link to it from the results page.

This on-the-fly search is superior to clustering proteins in the MiMI database based on sequence similarity, e.g. a HomoloGene-style [3] approach. A clustering is by definition a static object. Grouping MiMI proteins into static clusters and reporting interactions for the entire cluster would lose information relative to ranking molecules by their similarity to a given search.

When searching the extended MiMI database, BLAST is run against the set of distinct sequences which map to some MiMI molecule. This is 65,274 sequences comprising 66,513,740 residues. It takes approximately 4.5 seconds to search for a query sequence of 377 residues on a dual 2.0GHz Pentium IV Xeon with 3 gigabytes of RAM running Linux (kernel 2.6.12). It would be possible to precompute all the alignments; however, the storage space and computational requirements would be unreasonable given the short time to takes simply to run BLAST on the fly. However, if the MiMI database became an order of magnitude larger, precomputation on a large parallel cluster might become more attractive.

BLAST is fundamentally very easy to parallelize. The database can be partitioned horizontally and each query run against each partition, or the database can be replicated and the queries partitioned. The first solution can actually achieve a superlinear speedup - for each query, the database must be scanned, and if each partition fits in memory but the entire database does not then tremendous time savings can be achieved.

3. EVALUATION

Now that we have a way of returning additional interactions for any given molecule, we should consider ways to evaluate the results as well as the ranking of results. In this sense the search is more akin to the ranked list of results returned by a Web search engine than the set of results returned by a database query satisfying some Boolean predicate. In the database sense, a result is simply correct or incorrect. In the information retrieval sense, some results are more relevant than others, and we would like for the most relevant results to be ranked first.

If we have the evaluation results on some development set, we can attempt to develop a reranking strategy that weights the various features of each result (BLAST score, E value, percent of query aligned, percent of hit aligned, percent identity, percent conserved, number of gaps, etc) so that the final result ranking is more appropriate.

For each of the feasible evaluation strategies described below, informal experiments with a small set of common molecules were conducted to see if the evaluation strategy warranted further investigation. This development test set consisted of androgen receptor (AR), estrogen receptor (ER), tumor necrosis factor α (TNFα), and alcohol dehydrogenase.

3.1 Homology

One simple way to evaluate the results is simply to check whether each result is homologous to the original query. This is unsatisfactory for several reasons. First, we are not directly interested in homology - we are interested in interactions. Proteins may be homologous but only similar along part of their sequences, and thus unlikely to participate in many of the same interactions. In some cases it may not be clear whether we want to return a homologous molecule or not. For example, androgen receptor and progesterone receptor are both steroid receptors and share several domains, participate in many of the same interactions, yet are clearly not the same protein. So if we searched for an androgen receptor, we should first return all androgen receptors, then progesterone receptors, then perhaps other steroid receptors. Simply considering whether or not molecules are homologous does not capture this kind of situation. Also, although it is not difficult to scan result sets and determine which are homologous to the query to determine the percentage of results returned that are correct (that is, precision) it would be quite expensive to examine the entire database by hand and determine which proteins are homologous to the query to determine recall, that is, the percentage of correct results in the entire database that were returned. In fact, the only feasible way to determine this is to use the same tool (BLAST) with less stringent parameters to search the database for possibly homologous proteins. Also, all of the proteins returned using the stringent BLAST parameters above should be homologous, that is, one would expect virtually 100% precision. This is confirmed on the development test set. So, this evaluation strategy is not satisfactory.

3.2 Precision and Recall of Interactions

Another way to evaluate the results might be to evaluate precision and recall of the list of returned interactions. That is, collect the set of interactions for all returned molecules, and check both how many are true interactions for the query molecule, and how many of the true interactions for the query molecule are on the returned list. This is a more attractive evaluation strategy than precision and recall of homologous molecules, but is much more difficult to evaluate. For each pair of molecules, the only definitive way to determine if they truly interact is via experiment. Even if there was a list of interactions separate from the databases from which MiMI was constructed (for example [7] for androgen receptor), it may be the case that there are true interactions not on this list either. Also, interactions in the original source databases for MiMI may not always be correct, and the source databases may be missing interactions in the comprehensive list. If we used this evaluation strategy, we would need to compensate for errors in the original data. Thus to correctly evaluate precision and recall of interactions, compensating for errors in the original data, we need complete lists of all proteins the query and all hits interact with. If it were possible to obtain this data on a scale large enough for a reasonable evaluation, then we would not need MiMI or extended MiMI. Thus, evaluating precision and recall of interactions, while attractive in theory, is not feasible in practice.

3.3 Relevance Judgment

Another way to evaluate results is simply by asking an expert user to reorder results in the correct order by relevance,

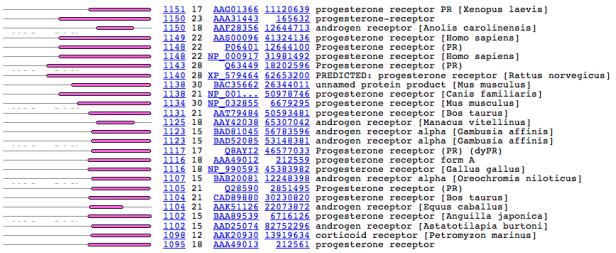

Figure 5: Portion of NCBI BLink results for GI 191936, *Mus musculus* androgen receptor.

or equivalently to assign a numerical relevance score to each returned result. One would expect relevance judgment to be more useful than homology based on the results of an NCBI BLink search for androgen receptor in which some other steroid receptors are ranked higher than some androgen receptors (Figure 3.3). However, whether because the MiMI database is smaller and has less noise, or because here the BLAST parameters are better optimized for finding highly related sequences, no similar examples could be found while testing the extended MiMI database. Since the deviations from the desired ranking appear to be minor, creating a test set for evaluation would probably not be especially informative unless the query proteins were selected randomly and a domain expert with intimate knowledge of the proteins carefully reranked them. This would likely be quite time-consuming and expensive. Also, creating development data for training a reranker did not seem warranted, since the BLAST score already seems to do an excellent job of ranking results by relevance. Nevertheless, as with test data, it is possible that a domain expert might be able to develop a useful training set for reranking, but this would again be quite expensive.

3.4 Extrinsic Evaluation

The final possibility would be to conduct an extrinsic evaluation, that is, evaluate the utility of the returned results for completing some other task. This is attractive since users do not use MiMI in isolation; they use it to help get their work done. However, it is not clear what the appropriate extrinsic evaluation would be, and a proper evaluation of this nature would be expensive and time-consuming. Thus, while an extrinsic evaluation is attractive, it is not practical within the scope of this study.

4. FUTURE WORK

4.1 Domain-Specific Interactions

The current version of MiMI simply reports interactions; it does not give any additional data as to why the two molecules interact. Given additional information of this nature, it would be possible to narrow the list of interactions returned for a given search based on whether the query and hit shared the property responsible for a given interaction. Typical properties might be the presence of a particular residue at a particular location on the protein, or the presence of a particular domain. (Continuing the programming analogy from earlier, a domain is roughly analogous to a library function - a common sequence that appears in dif-

ferent proteins. One common domain is zinc finger, which) Annotations of many proteins are available in the NCBI databases as well as other databases; however, little information about which features are responsible for which interactions is currently available. An interesting experiment might be to attempt to induce hypotheses for why two particular molecules interact using the MiMI data and an EM-style algorithm. This algorithm would predict features of the proteins likely to be responsible for the observed interactions; the predicted features would be automatically chosen to maximize the likelihood of the observed data. This has been tried on various smaller data sets with some success, e.g. [11] on PFAM.

4.2 Other Applications

This general normalization approach would be interesting to apply to other domains. One interesting application would be music databases. There are at least two large databases of traditional music on the Internet: thesession.org and JC's ABC Tune Finder [9]. Both search music in ABC format, which is not a music format like MP3 but instead a semantic representation of the tune, much like sheet music. thesession.org allows searching by tune name or by fragment of tune, but the fragment must be an exact match. JC's ABC tune finder allows more advanced searching, e.g. by contour (the pattern of ascending or descending notes). This provides some rudimentary similarity searching, but it would be interesting to implement a BLAST-style algorithm to provide full-blown similarity searching complete with score matrix, be it on the actual notes or some simplified coded representation thereof.

5. CONCLUSION

Although I have not formally evaluated the extended MiMI database, I have constructed what I believe to be a useful complement to the original MiMI. By finding the connected components of the graph induced by equivalence of external IDs, a larger list of names and external IDs for each MiMI record is a now available. Given a MiMI molecule, users can find identical and highly similar molecules. Also, users can search for any protein in the much larger RefSeq and NR databases as well as for arbitrary protein sequences. I have also presented several possibilities for more formal evaluations that could be conducted with more time or money, and expect that the extended MiMI database would compare favorably to the original MiMI database in terms of usefulness to researchers in the biomedical field. The system is currently available at http://oriole.eecs.umich.edu/mimi.

6. ACKNOWLEDGEMENTS

Thanks to Dr. David J. States and Carlos Santos for suggesting this problem and helpful discussion and to Magesh Jayapandian and Dr. H.V. Jagadish for access to the MiMI database.

7. REFERENCES

[1] Sequence identifiers: A historical note. `http://www.ncbi.nlm.nih.gov/Sitemap/sequenceIDs.html`, June 2004.

[2] Blink. `http://www.ncbi.nlm.nih.gov/sutils/static/blinkhelp.html`, May 2005.

[3] Homologene. `http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=homologene`, November 2005.

[4] Bruce Alberts, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts, and Peter Walter. *Molecular Biology of the Cell (4th ed.)*. Garland Publishing, New York, 2002.

[5] SF. Altschul, TL. Madden, AA. Schaffer, J. Zhang, Z. Zhang, W. Miller, and DJ. Lipman. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic Acids Research*, 25(17):3389–3402, September 1997.

[6] Joseph Bedell, Ian Korf, and Mark Yandell. *BLAST*. O'Reilly, July 2003.

[7] LK Beitel. Androgen receptor-interacting proteins. `http://www.androgendb.mcgill.ca/ARinteract.pdf`, March 2002.

[8] Bobby-Joe Breitkreutz, Chris Stark, and Mike Tyers. The grid: The general repository for interaction datasets. *Genome Biology*, 4(3), 2003.

[9] John Chambers. Jc's abc tune finder. `http://trillian.mit.edu/~jc/music/abc/findtune.html`.

[10] Adriane Chapman, Cong Yu, and H.V. Jagadish. Effective integration of protein data through better data modeling. *OMICS: A Journal of Integrative Biology*, 7(1):101–102, July 2003.

[11] Minghua Deng, Shipra Mehta, Fengzhu Sun, and Ting Chen. Inferring domain-domain interactions from protein-protein interactions. *Genome Research*, 12(10):1540–1548, October 2002.

[12] C. Alfarano et al. The biomolecular interaction network database and related tools 2005 update. *Nucleic Acids Research*, 33(Database issue):D418–D424, 2005.

[13] S. Peri et al. Human protein reference database as a discovery resource for proteomics. *Nucleic Acids Research*, 32(Database issue):D497–D501, 2004.

[14] S. Henikoff and J.G. Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*, 89:10915–10919.

[15] Danielle Kemmer, Yong Huang, Sohrab P Shah, Jonathan Lim, Jochen Brumm, Macaire MS Yuen, John Ling, Tao Xu, Wyeth W Wasserman, , and BF Francis Ouellette. Ulysses - an application for the projection of molecular interactions across species. *Genome Biology*, 6(12), 2005.

[16] KD Pruitt, T Tatusova, and DR Maglott. Ncbi reference sequence (refseq): a curated non-redundant sequence database of genomes, transcripts and proteins.

[17] L. Salwinski, C.S. Miller, A.J. Smith, F.K. Pettit, J.U. Bowie, and D. Eisenberg. The database of interacting proteins: 2004 update. *Nucleic Acids Research*, 32(Database issue):D449–D551, 2004.

[18] SP. Shah, Y. Huang, T. Xu, MM. Yuen, J. Ling, and BF. Ouellette. Atlas - a data warehouse for integrative bioinformatics. *BMC Bioinformatics*, 6(1):34, February 2005.

[19] TL. Shi, YX. Li, YD. Cai, and KC. Chou. Computational methods for protein-protein interaction and their application. *CURRENT PROTEIN & PEPTIDE SCIENCE*.