

Biomolecular annotation prediction through information integration

Davide Chicco, Marco Tagliasacchi, Marco Masseroli
Dipartimento di Elettronica e Informazione, Politecnico di Milano,
Piazza Leonardo Da Vinci 32, 20133 Milano, Italy
Email: davide.chicco@elet.polimi.it, marco.tagliasacchi@polimi.it,
marco.masseroli@polimi.it

Keywords: Biomolecular databases, Bioinformatics data integration, Biomolecular annotation prediction, Information integration, Data warehouse, Software infrastructures.

Abstract. In the recent years, an increasingly large amount of biomedical and biomolecular information and data has become available to researchers, allowing to the scientific community to infer new knowledge and reach new objectives. As these information increase, so does the difficulty in managing it efficiently. In this paper, we present a short overview of our proposal to solve this problem, a prototypal multi-organism Genomic and Proteomic Data Warehouse called GPDW, based at Politecnico di Milano. We also present the computational methods we implemented to exploit it. Experimental studies on datasets demonstrated the effectiveness of our resource and methods.

1 Introduction

The amount of biomedical and biomolecular information and data has grown in recent years, presenting new challenges and goals in the bioinformatics scientific community. Such large amounts of information makes computerized integration and analysis of these data unavoidable [1, 2], in particular controlled annotations of biomolecular entities (mainly genes and their protein products) are of great value to support scientists and researchers.

These data can effectively support the biomedical interpretation of biomolecular test results and the extraction of knowledge; which can then be used to formulate and validate hypotheses, and possibly discover new biomedical knowledge. Several computational tools, such as those for enrichment analysis [3, 4, 5, 6, 7], are available to analyze such annotations.

The information is often present in multiple resources, in different locations and formats, and within different technologies and electronic infrastructures; so, the integration of these data, in order to infer new knowledge, can be very difficult [8, 9].

Different databank integration approaches exist. In order to build an integrated repository of distributed valuable biomolecular information and use it to support innovative knowledge discovery methods, for several reasons we decided to follow the *data warehousing* approach. The main motivation for this is that having all the data together allows a quicker, more efficient and effective access to the data. The "on demand" approaches, instead, are slower and may only be effective for specific or singular queries. Our approach comprises two different parts: the first is *integration* of data from different sources; and, secondly, *analysis* of the integrated data, with computational methods, to infer new knowledge.

Given integration architecture able to manage and integrate a large amount of information, the computational methods and tools needed to analyze the information are manifold, especially concerning controlled annotations (e.g. Gene Ontology (GO) annotations [10]). In this scenario, we propose a method for predicting new annotations of gene and gene products based solely on previously available annotations. Our first

contribution extends and enhance the method in [11], hereafter denoted the SVD (*Singular Value Decomposition*) method, by incorporating a gene (or gene product) clustering algorithm based on the functional similarity between gene (or gene product) pairs. The proposed method, denoted the SIM (*Semantic IMprovement*) method, computes a separate set of eigen-terms for each identified cluster, while the original SVD method computes a global set of eigen-terms.

This paper is organized as follow: in Section 2, we will explain our data warehousing approach and information integration methods; in Section 3, we will briefly describe the computational methods we use to elaborate the integrated data; in Section 4, we will discuss our software implementation choices. Finally conclusions and possible future developments are drawn in Section 5.

2 Data warehousing and information integration

In this section we will explain about our prototypal multi-organism Genomic and Proteomic Data Warehouse, called GPDW, based at Politecnico di Milano (Subsection 2.1), and we will describe its information and data integration approach (Subsection 2.2).

2.1 Genomic and Proteomic Data Warehouse

Given the large amount of biomolecular information and data available nowadays in different web sources, information integration has become a very important task for the bioinformatics community. There are different data integration approaches available:

- *federated databases* (e.g. BioKleisli [12], K2 [13]);
- *link driven federations* (e.g. SRS [14]);
- *link-driven integration of data sources* (e.g. Entrez [15], GeneCards [16], SOURCE [17]);
- *mediator-based architecture* (e.g. BioDataServer [18], DiscoveryLink [19]);
- *data warehousing* (e.g. EnsMart [20], BioWarehouse [21]).

In order to integrate heterogeneous bioinformatics data available in a variety of formats from many different sources, we decided to follow this last approach, *data warehousing*. This is because, generally, accessing to data warehouse systems is quicker, and more efficient and effective than accessing the "on demand" systems. When used for singular and specific queries, "on demand" approaches could be preferable. In our case, we decided to divide our system into two parts: an *integration* part, where all the data sources are put together; and an *analysis* part, where all the data are used to infer new knowledge.

Our objective was to generate, maintain updated and extend a multi-species Genomic and Proteomic Data Warehouse (GPDW) that provided transparent provenance tracking, quality checking of all integrated data, comprehensive annotation based analysis support, identification of unexpected information patterns and which fostered biomedical discoveries. This data warehouse constitutes the back-end of a system that we will exploit through suitable web services, and will allow discovery of new knowledge by analyzing our large amount of information and data available.

2.2 Information and data integration approach

The integration of data from different sources in our data warehouse is managed in three phases: *importing*; *aggregation* and *integration*. In the former (*importing*), data from the different sources are imported into one repository. In the second (*aggregation*), they are gathered and normalized into a single representation in the instance-aggregation

tier of our global data model. In the third (*integration*), data are organized into informative clusters in the concept-integration tier of the global model.

During the initial importing and aggregation phases, tables of the features described by the imported data are created and populated. Then, similarity and historical ID (*IDentification*) data are created by translating the IDs provided by the data sources to our internal OIDs (*Own IDentification*).

In doing so, relationship data are coupled with their related feature entries. According to the imported data sources and their mutual synchronization, relationship data may refer to feature entries or features that have not been imported into the data warehouse. In this case, missing integrated feature entries are synthesized and labeled as such (i.e., inferred through synthesis from relationship data). However, if a missing entry has an obsolete ID and through unfolded translated historical data it is possible to get a more current ID for it, the relationship is first moved to the latest ID and is marked as *inferred* through historical data. In this way, all the relationships expressed by the imported relationship data are preserved and allows their subsequent use to infer new biomedical knowledge discoveries (e.g., by transitive closure inference, also involving the synthesized entries). The final integration phase uses similarity analysis to test whether single feature instances from different sources represent the same feature concepts. In this case, they are associated with a new single concept OID. To keep track of the inference method used to derive an entry, an *inference* field is used in all tables that have been integrated. Furthermore, a summary quality rate for each concept instance is computed based on the source-specific instances contributing to the concept instance and its *inferred* attribute value.

At the end of the integration process, the defined indexes, unique, primary and foreign key integrity constraints of all the integrated tables are enforced in order to detect and resolve possible data inconsistencies and duplications, thus improving the speed of access to the integrated data, as well as their general quality.

3 Computational methods

In this section we describe how to exploit the integrated data integrated in the data warehouse in order to generate good predictions of new biomolecular annotations.

3.1 Prediction of biomolecular annotations

Genome sequencing has completely revolutionized the approach of studying biological functionalities. To better understand the functions of genes and proteins, the concept of annotation (association of nucleotide or amino acid sequences with useful information) was developed. This information is expressed through controlled vocabularies, sometimes structured as ontologies, where every controlled term of the vocabulary is associated with a unique alphanumeric code. Such a code associated with a gene or protein ID constitutes an annotation. The annotation curator's task is paramount for the correct use of the annotations. Curators discover and/or validate new annotations; and publish them in online web databanks, thereby making them available to scientists and researchers. However, available annotations are incomplete, and, for recently studied genomes, they are mostly computationally derived. So only a few of them represent highly reliable human-curated information. To support and quicken the time-consuming curation process, prioritized lists of computationally predicted annotations are extremely useful. In our work, we assessed, improved and extended a prediction method already available in literature, based on SVD (*Singular Value Decomposition*) [11] of the annotation matrix, by proposing a new method, SIM (*Semantic IMprovement*). A flow chart describing SVD and SIM methods is provided in Figure 1.

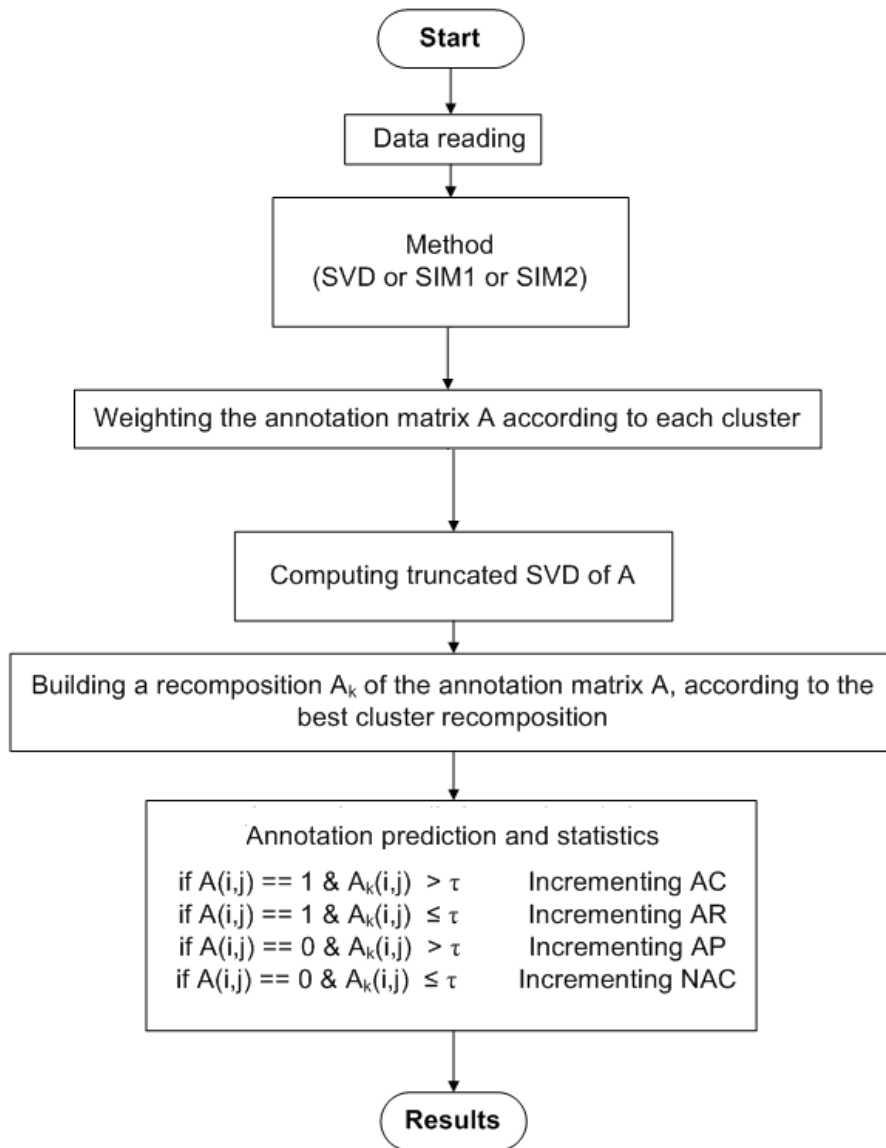


Figure 1: The flow chart describing SVD and SIM methods.

3.2 SVD - Singular Value Decomposition

Let the matrix $A \in \{-1, 0, 1\}^{m \times n}$, with m rows corresponding to genes and n columns corresponding to annotation terms, represent all the considered annotations of controlled vocabularies for a certain organism. The entry $A(i, j)$ assumes a value of 1 if gene i is annotated to term j or to any descendant of j in the considered ontology structure. It assumes a value of -1 if it is known that gene i must not be annotated to term j , or a value of 0 otherwise. Then, we considered the SVD of the annotation matrix A , i.e. $A = U\Sigma V^T$, where U is a $m \times p$ unitary matrix (i.e. $U^T U = I$), Σ is a non-negative diagonal matrix of size $p \times p$, and V is a $n \times p$ unitary matrix, where $p = \min(m, n)$. An annotation prediction is performed based on this SVD is performed by computing a reduced rank approximation A_k of matrix A by means of singular value decomposition (where $0 < k < r$, with r the number of non zero singular values). A_k contains real valued entries related to the likelihood that gene i shall be annotated to term j . For a defined threshold τ , if $A_k(i, j) > \tau$, gene i is predicted to be annotated to term j . The threshold τ can be chosen in order to obtain the B best predicted annotations (with $B \in \mathbb{N}$). Values of k and τ can be heuristically chosen according to preliminary tests on the specific data considered.

3.3 SIM - Semantic Improvement

The SVD method implicitly adopts a global term-to-term correlation matrix $T = A^T A$, estimated from the whole set of available annotations. In contrast, we propose an adaptive approach, called the SIM method, which clusters genes (the rows of matrix A) based on their original annotation profile and values a set of distinct correlation matrices, T_c , one for each cluster. For each matrix, T_c , a predicted annotation profile for gene i is computed. The selected predicted annotation profile for gene i is the one that minimizes variation, measured by the ell-2 norm, with respect to the original annotation profile of the gene.

We heuristically fix a number C of clusters, and completely discard the columns of matrix U where $j = C+1, \dots, n$. Each column, u_c , of SVD matrix U represents a cluster, and the value $U(i, c)$ indicates the membership of gene i to the c -th cluster [22]. We use this membership degree to cluster the genes (the rows of matrix A). As such, each gene might belong to more than one cluster with different degrees of membership. We notice that the columns of U are a set of eigenvectors for the matrix $G = AA^T$, i.e. the similarity between gene pairs is measured by the inner product of their annotation profiles. To estimate the correlation matrices, T_c , we cluster genes based on their functional similarity, expressed through their annotations, by exploiting SVD of matrix A . To calculate T_c , for each cluster, first we generate a modified gene-to-term matrix $A_c = W_c A$ (where $W_c \in \mathbb{R}^{m \times m}$ is a diagonal matrix with the entries of u_c along the main diagonal), in which the i -th row of A is weighted by the membership score of the corresponding gene to the c -cluster. Then, we compute $T_c = A_c^T A_c$.

Furthermore, to effect more accurate clustering, we compute the eigenvectors of the matrix $\tilde{G} = A S A^T$ where $S \in \mathbb{R}^{n \times n}$ is the term similarity matrix. Starting from a pair of ontology terms, j_1 and j_2 , the term functional similarity $S(j_1, j_2)$ can be calculated using different methods; here we use the Lin's similarity metrics [23].

An important and interesting difference between our method and that of Draghici et al. [11] is the choice of the predicted annotations. While Draghici et al. [11] took all the predicted annotations above a certain threshold τ as equally correct, our method provides the predicted annotations in order of accuracy. Thus, the user can select the B most correct annotations by choosing the value of τ threshold.

To assess the performance of the SVD and SIM methods, we considered the Gene Ontology annotations of different organisms, including *Saccharomyces cerevisiae* and *Drosophila melanogaster*, excluding annotations with evidence code IEA (*Inferred Elec-*

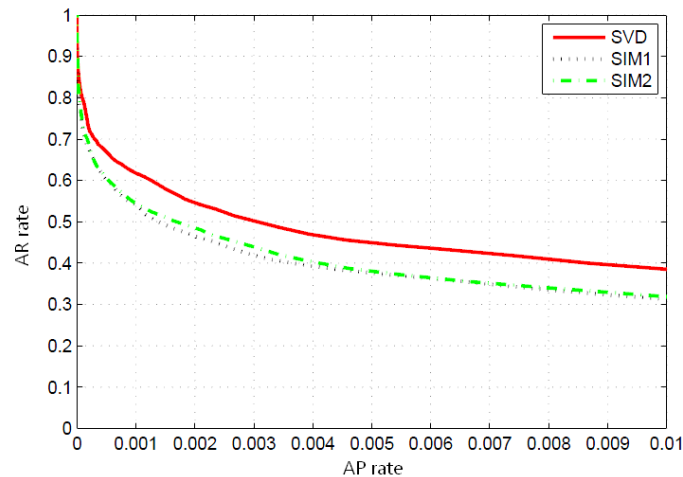


Figure 2: Annotations to be reviewed (AR) rate versus predicted annotations (AP) rate for the prediction of the Gene Ontology Biological Process annotations of *Saccharomyces cerevisiae* (SGD) genes.

tronic Annotations), since they have not been checked by a manual curator. After this, the *Saccharomyces cerevisiae* dataset included 3,676 genes related to 1,256 ontology terms, for a total of 23,384 annotations. For the sake of brevity in this paper, we present only the result for the *Saccharomyces cerevisiae* dataset, but similar results were obtained also for the *Drosophila melanogaster* dataset.

3.4 Results

We assessed the performance of the SVD and SIM methods by performing K-fold cross-validation as in [24] and [25], and discarding terms used to annotate less than M genes, in order to avoid considering very low reference annotation terms which could bias the evaluation. As for the SIM method, we evaluated two variants. In SIM1, we set $S = I$, i.e. the clustering step does not rely on the functional similarity between terms. In SIM2, matrix S is computed by means of the Lin's metrics [23]. In both cases we heuristically set a fixed number of clusters $C = 5$ for all ontology, based on our own tests ($C \ll p$). With a threshold value τ , if $A(i, j) \leq 0$ and $A_k(i, j) > \tau$, a new annotation is suggested; this case is denoted as a *predicted annotation* (AP). Conversely, if $A(i, j) = 1$ and $A_k(i, j) \leq \tau$, an existing annotation is suggested to be semantic inconsistent with the available data; this case is denoted as an *annotation to be reviewed* (AR). The *confirmed annotations* (AC) rate and *no annotations confirmed* (NAC) rate are similarly defined.

The curve in Figure 2 depicts the trade-off between the annotations to be reviewed rate (AR / (AC + AR)) and predicted annotations rate (AP / (AP + NAC)) of annotations of *Saccharomyces cerevisiae* (SGD) genes, when the prediction is performed using heuristically the first $k = 40$ eigenvectors of the available annotation matrix, A_k . To improve reliability of validation, we performed several validations with different A_k values by retaining Gene Ontology terms used to annotate at least $M = 3$ or $M = 10$ genes of the considered organism and excluding annotations with evidence code IEA (inferred electronic annotations), since they have not been checked by a manual curator.

As an aggregated indicator of the prediction performance, we computed the area above the AR rate versus AP rate curve (AAC) in the [0;0:01] range, for *Saccharomyces cerevisiae* (SGD). Indeed, we are typically interested in the low range of AP rate, since it corresponds to top-ranked predictions of newly inferred annotations (AP) with the highest score. The normalized AAC metrics are limited to the [0;1] interval, where a value closer to 1 implies more accurate predictions. In all cases, we computed the AAC metrics considering only the prediction of Gene Ontology terms with depth L from the

Table 1: Area above the curve of annotations to be reviewed (AR) rate versus predicted annotations (AP) rate of the Gene Ontology annotations of *Saccharomyces cerevisiae* (SGD), at Gene Ontology level greater than 2 ($L > 2$) and 6 ($L > 6$) predicted with different methods when only terms annotating at least M genes are considered for prediction. BP: biological processes, MF: molecular functions, CC: cellular components GO ontologies; k : number of eigenvectors of the considered annotation matrix retained for prediction. Testing values 20 and 40 for k were heuristically chosen as sample values based on preliminary tests made.

M	method	k	BP		MF		CC	
			L > 2	L > 6	L > 2	L > 6	L > 2	L > 6
3	SVD	20	0.58	0.35	0.47	0.51	0.39	0.50
		40	0.65	0.57	0.57	0.60	0.32	0.51
	SIM1	20	0.64	0.53	0.56	0.64	0.41	0.60
		40	0.70	0.61	0.52	0.61	0.37	0.56
	SIM2	20	0.64	0.50	0.59	0.70	0.37	0.56
		40	0.71	0.62	0.55	0.62	0.36	0.60
10	SVD	20	0.53	0.34	0.43	0.49	0.35	0.43
		40	0.60	0.53	0.53	0.59	0.31	0.47
	SIM1	20	0.62	0.52	0.50	0.56	0.43	0.56
		40	0.65	0.60	0.46	0.39	0.39	0.56
	SIM2	20	0.63	0.52	0.54	0.65	0.37	0.53
		40	0.67	0.58	0.49	0.47	0.35	0.56

root of the ontology greater than either 2 or 6. Using the AAC metrics, Table 1 shows that the SIM method outperforms the SVD method for all Gene Ontology ontologies. In 66% of cases SIM2 was better than, or equal to, the SIM1 method, showing that clustering based on the functional similarity between terms might be beneficial. Nevertheless, most of the performance gain between SIM and SVD stems from the adaptive nature of SIM, regardless of how clustering is actually performed. In fact, the SVD method, which computes similarities between clusters in terms of frequency of co-annotation, is bound to be biased towards the larger clusters, since it is unnormalized. The SIM method counterbalances such a bias with its adaptive approach of clustering genes (or gene products) according to their original annotation profile. A similar analysis was conducted on the Gene Ontology annotations of other organisms showing comparable results.

The methods we proposed turned out to be very useful and powerful compared to those already present in literature. Furthermore, since our approach is not limited to the Gene Ontology and can be applied to any controlled annotation, increasingly available multiple annotations of genes and gene products from different controlled vocabularies and ontologies could be jointly considered to further improve prediction reliability.

4 Software infrastructure

Since the amount of data was very large and the objectives and computation quite demanding, we had to pay particular attention to software performance and memory usage. To satisfy the facility of the software to be modified and extended, we first chose the Java programming language for implementation. Java results to be very independent from platform and from operating system, that is a very important value for the software objectives. However, Java shows some limitations, too: it provides a high response time, and uses a lot of memory. The response delay clearly results you comparing matrix operation rapidity in Java and C++ environments with native solutions that use very highly optimized mathematical kernels. The high memory usage problem is especially relevant

for the virtual machine.

Given these issues, we decided to implement the mathematical core of our software in C++ programming language using a multiplatform, multithreading, optimized mathematical kernel, such as AMD Core Math Library (ACML) [26]. This library provides a high level of optimization on generic processors, and is simple to use, given that it does not need to be compiled. In addition, ACML is freely available for both Linux/Unix, Microsoft Windows and Solaris systems. Another effective library we used for the mathematical core was SVDLIBC [27]. The multithreading native part was developed by using OpenMP (Open Multi-Processing, OPM) [28] compiler directives which exploited by the mathematical kernels, independently from the operating system. The interaction between the native C++ code and Java code was through Java Native Interface (JNI) [29].

4.1 ACML and SVDLIBC

We needed a way to implement a complex data processing efficiently, and we chose ACML and SVDLIBC libraries to manage this issue. The AMD Core Math Library (ACML) implements a set of multithread functions optimized for *high performance computing*. It is formed by the following main components, relevant to our implementation:

- Complete implementation of the Basic Linear Algebra Subprograms (BLAS) [30]. BLAS is a *standard* application programming interface, supported by most of the mathematical libraries to execute vector and matrix operations. Implemented in different libraries, developed both by open source communities (GotoBLAS, ATLAS, etc) and processor inventors (ACML, Intel's MKL, IBM's ESSL, etc), BLAS provides different operators, mainly for: operations between vectors, between vectors and matrices, and between matrices.
- Complete implementation of the Linear Algebra PACKage (LAPACK) [32]. LAPACK is a set of functions, written in Fortran to make high level scientific calculations. It is mainly used to solve: linear simultaneous equations, systems with linear least squares solutions, factoring problems, and for eigenvalue and eigenvector investigations.

Moreover, the ACML could be compiled for 32-bit and 64-bit systems, on Linux/Unix, Microsoft Windows or Solaris systems. ACML takes great advantage of OpenMP [28] resources; this allows implementation with simple threading models and a simplified debug. OpenMP also permits multithreading usage on any system.

SVDLIBC is a C library based on the SVDPACKC library [31], used mostly for SVD and truncated SVD operations. Its intent is to provide an easy to use interface and a set of functionalities to manage, and convert matrices. The SVDPACKC algorithm used in the software is *las2*, which proved to be very powerful for the SVD calculation despite of having some inaccuracies in the minor eigenvalues calculation.

5 Conclusions

Easy and integrated access to the high amount of biomolecular information and knowledge now available in many heterogeneous and distributed data sources is required to answer biological questions. Data warehousing and computational systems can provide support for comprehensive use and analysis of sparsely available genomic and proteomic structural, functional and phenotypic information and knowledge. We designed a method for integrating biomolecular knowledge (mainly expressed through controlled terminologies or ontologies), into a data warehouse, and exploiting such integrated data to predict new biomolecular annotations.

We proposed a novel contribution in the context of prediction of genomic ontological

annotations, SIM. Our Semantic IMproved (SIM) version of the Single Value Decomposition (SVD) method produced better predictions than the SVD method alone. In the future, our software implementation could be improved in many ways. For example, an algorithm for the automatic computation of k , τ and C parameters based on Receiver operating characteristic (ROC) curves [34] could be added. Furthermore, since our approach is not limited to a specific type of annotations, can be applied to any controlled annotation. Increasingly available multiple annotations of genes and gene products from different ontologies and controlled vocabularies could be jointly considered to further improve prediction reliability.

Finally, our aim is to provide web service access to our implemented method and integrate such web service with other available services within the Search Computing framework (<http://www.search-computing.com>) [35] in order to provide support for answering complex life science questions [36].

Acknowledgments

This research is part of the Search Computing project (2008-2013) funded by the European Research Council (ERC), IDEAS Advanced Grant.

References

- [1] M.Y. Galperin, and G.R. Cochrane, "Nucleic Acids Research Annual Database Issue and the NAR Online Molecular Biology Database Collection in 2009". *Nucleic Acids Res.*, vol. 37, no. Database issue, pp. D1–D4, 2009.
- [2] EMBL Nucleotide Sequence Database Statistics, <http://www3.ebi.ac.uk/Services/DBStats/>, 2009
- [3] D.W. Huang, B.T. Sherman, and R.A. Lempicki, "Bioinformatics Enrichment Tools: Paths toward the Comprehensive Functional Analysis of Large Gene Lists". *Nucleic Acids Res.*, vol. 37, no. 1, pp. 1–13, 2009.
- [4] F. Al-Shahrour, P. Minguez, J. Trraga, I. Medina, E. Alloza, D. Montaner, and J. Dopazo, "FatiGO+: A Functional Profiling Tool for Genomic Data. Integration of Functional Annotation, Regulatory Motifs and Interaction Data with Microarray Experiments". *Nucleic Acids Res.*, vol. 35, no. Web Server issue, pp. W91–W96, 2007.
- [5] D.W. Huang, B.T. Sherman, Q. Tan, J. Kir, D. Liu, D. Bryant, Y. Guo, et al., "DAVID Bioinformatics Resources: Expanded Annotation Database and Novel Algorithms to Better Extract Biology from Large Gene Lists". *Nucleic Acids Res.*, vol. 35, no. Web Server issue, pp. W169–W175, 2007.
- [6] M. Masseroli, D. Martucci, and F. Pincioli, "GFINDER: Genome Function INtegrated Discoverer through Dynamic Annotation, Statistical Analysis, and Mining". *Nucleic Acids Res.*, vol. 32, pp. W293–W300, 2004.
- [7] M. Masseroli, "Management and Analysis of Genomic Functional and Phenotypic Controlled Annotations to Support Biomedical Investigation and Practice". *IEEE Trans. Inf. Technol. Biomed.* vol. 11, no. 4, pp. 376–385, 2007.
- [8] W. Sujansky, "Heterogeneous Database Integration in Biomedicine". *J. Biomed. Inform.*, vol. 34, no. 4, pp. 285–298, 2001.
- [9] T. Hernandez, and S. Kambhampati, "Integration of Biological Sources: Current Systems and Challenges ahead". *SIGMOD Record*, vol. 33, no. 3, pp. 51–60, 2004.
- [10] The Gene Ontology Consortium, "Creating the gene ontology resource: Design and implementation". *Genome Res.*, vol. 11, pp. 1425–1433, 2001.
- [11] P. Khatri, B. Done, A. Rao, A. Done, and S. Draghici, "A semantic analysis of the annotations of the human genome". *Bioinformatics*, vol. 21, no. 16, pp. 3416–3421, 2005.
- [12] S.B. Davidson, C. Overton, V. Tanen, and L. Wong, "BioKleisli: A Digital Library for Biomedical Researchers". *Int. J. Digit. Libr.*, vol. 1, pp. 36–53, 1997.
- [13] S.B. Davidson, J. Crabtree, B. Brunk, J. Schug, V. Tannen, C. Overton, and C. Stoeckert, "K2/Kleisli and GUS: Experiments in Integrated Access to Genomic Data Sources". *IBM System Journal*, vol. 40, no. 2, pp. 512–531, 2001.
- [14] T. Etzold, A. Ulyanov, and P. Argos, "SRS: Information Retrieval System for Molecular Biology Data Banks". *Methods Enzymol.*, vol. 266, pp. 114–128, 1996.
- [15] T.A. Tatusova, I. Karsch-Mizrachi, and J.A. Ostell, "Complete Genomes in WWW Entrez: Data Representation and Analysis". *Bioinformatics*, vol. 15, pp. 536–543, 1999.
- [16] M. Safran, I. Solomon, O. Shmueli, M. Lapidot, S. Shen-Orr, A. Adato, et al., "GeneCards 2002: Towards a Complete, Object-Oriented, Human Gene Compendium". *Bioinformatics*, vol. 18, no. 11, pp. 1542–1543, 2002.

- [17] M. Diehn, G. Sherlock, G. Binkley, H. Jin, J.C. Matese, T. Hernandez-Boussard, et al., "SOURCE: A Unified Genomic Resource of Functional Annotations, Ontologies, and Gene Expression Data". *Nucleic Acids Res.*, vol. 31, pp. 219–223, 2003.
- [18] A. Freier, R. Hofestdt, M. Lange, U. Scholz, and A. Stephanik, "BioDataServer: A SQL-Based Service for the Online Integration of Life Science Data". *In Silico Biol.*, vol. 2, no. 2, pp. 37–57, 2002.
- [19] L.M. Haas, P.M. Schwarz, P. Kodali, E. Kotlar, J.E. Rice, and W.C. Swops, "DiscoveryLink: A System for Integrated Access to Life Sciences Data Sources". *IBM Systems Journal*, vol. 40, no. 2, pp. 489–511, 2001.
- [20] A. Kasprzyk, D. Keefe, D. Smedley, D. London, W. Spooner, C. Melsopp, et al., "EnsMart: A Generic System for Fast and Flexible Access to Biological Data". *Genome Res.*, vol. 14, no. 1, pp. 160–169, 2004.
- [21] T.J. Lee, Y. Pouliot, V. Wagner, P. Gupta, D.W. Stringer-Calvert, J.D. Tenenbaum, and P.D. Karp, "BioWarehouse: A Bioinformatics Database Warehouse Toolkit". *BMC Bioinformatics*, vol. 7, no. 170, pp. 1–14, 2006.
- [22] P. Drineas, "Clustering large graphs via the singular value decomposition: Theoretical advances in data clustering". *Machine Learning*, vol. 56, pp. 9–33, July 2004.
- [23] D. Lin, "An Information-Theoretic Definition of Similarity". *Proceedings of the Fifteenth International Conference on Machine Learning (ICML '98)*, Jude W. Shavlik (Ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp 296–304, 1998.
- [24] O. D. King, R. E. Foulger, S. S. Dwight, J. V. White, and F. P. Roth, "Predicting gene function from patterns of annotation". *Genome Res.*, vol. 13, no. 5, pp. 896–904, 2003.
- [25] Y. Tao, L. Sam, J. Li, C. Friedman, and Y. A. Lussier, "Information theory applied to the sparse gene ontology annotation network to predict novel gene function". *Bioinformatics*, vol. 23, no. 13, pp. 529–538, 2007.
- [26] AMD Core Math Library (ACML), <http://developer.amd.com/cpu/libraries/acml/>, 2002.
- [27] D.Rohde, SVDLIBC, <http://tedlab.mit.edu/~dr/SVDLIBC>, 2004.
- [28] L. Dagum, R. Menon, "OpenMP: an industry standard API for shared-memory programming". *IEEE Computational Science & Engineering*, vol. 5, issue 1, pp. 46–55, 1998.
- [29] R. Gordon, "Essential JNI: Java Native Interface". *Prentice-Hall, Inc.*, NJ, USA, 1998.
- [30] C.L. Lawson, R.J. Hanson, D.R. Kincaid, F.T. Krogh, "Basic Linear Algebra Subprograms for Fortran Usage". *ACM Transactions on Mathematical Software (TOMS)*, vol. 5, issue 3, 1979.
- [31] M. Berry, T. Do, G. O'Brien, V. Krishna, S. Varadhan, "SVDPACKC (Version 1.0) User's Guide". *Citeseer*, 1993.
- [32] B. Angerson, G. Dongarra, D.C. McKenney et al., "LAPACK: A portable linear algebra library for high-performance computers". *Proceedings of the 1990 ACM/IEEE conference on Supercomputing*, pp.2–11, 1990.
- [33] T. Hofmann, "Probabilistic Latent Semantic Indexing". *Proceedings of the Twenty-Second Annual International SIGIR Conference on Research and Development in Information Retrieval (SIGIR-99)*, 1999.
- [34] J.P. Egan, "Signal Detection Theory and ROC Analysis". *Academic Press*, New York, 1975.
- [35] S. Ceri, M. Brambilla, "Search Computing - Challenges and Directions". *Springer, Lecture Notes in Computer Science*, vol. 5950, 2010.
- [36] M. Masseroli, G. Ghisalberty, "Bio-SeCo: Integration and Global Ranking of Biomedical Search Results". *Springer, Lecture Notes in Computer Science*, vol. 6585, pp. 203–214, 2011.