Software Suite for Gene and Protein Annotation Prediction and Similarity Search

Davide Chicco, *student member, IEEE,* and Marco Masseroli

Abstract—In the computational biology community, machine learning algorithms are key instruments for many applications, including the prediction of gene-functions based upon the available biomolecular annotations. Additionally, they may also be employed to compute similarity between genes or proteins. Here, we describe and discuss a software suite we developed to implement and make publicly available some of such prediction methods and a computational technique based upon Latent Semantic Indexing (LSI), which leverages both inferred and available annotations to search for semantically similar genes. The suite consists of three components. *BioAnnotationPredictor* is a computational software module to predict new gene-functions based upon Singular Value Decomposition of available annotations. *SimilBio* is a Web module that leverages annotations available or predicted by *BioAnnotationPredictor* to discover similarities between genes via LSI. The suite includes also *SemSim*, a new Web service built upon these modules to allow accessing them programmatically. We integrated *SemSim* in the Bio Search Computing framework (http://www.bioinformatics.deib.polimi.it/bio-seco/seco/), where users can exploit the Search Computing technology to run multi-topic complex queries on multiple integrated Web services. Accordingly, researchers may obtain ranked answers involving the computation of the functional similarity between genes in support of biomedical knowledge discovery.

Index terms— Latent Semantic Indexing, Singular Value Decomposition, gene similarity search, biomolecular annotations, Gene Ontology, Web service, semantic similarity, Search Computing

1 Introduction

Controlled biomolecular annotations are paramount to describe biomolecular knowledge and support biomedical investigation. They consist of associations between biomolecular entities (genes or proteins) and controlled terms that describe the biomolecular entity features or functions; these terms are often part of an ontology, i.e. they are related through semantic relationships that allow their use for inferential analyses. Despite their importance, available controlled annotations suffer from incompleteness and the presence of errors. In this context, computational methods that apply efficient machine learning and data mining algorithms to predict missing annotations, or suggest available annotations to be revised (both ranked by their likelihood) are of paramount importance in the field [1]. Additionally, these methods leverage available and predicted annotations to support semantic similarity search of biomolecular entities. Using advanced computational techniques, based upon Singular Value Decomposition (SVD) [2], we developed some software components to predict biomolecular annotations and compute semantic similarity between biomolecular entities; here, we illustrate and discuss them.

We started with the approach developed by Khatri and colleagues [3], which is based upon truncated Singular Value Decomposition (tSVD) [2]. In our previous work, we enhanced it, by defining an automatic method to choose the SVD truncation level built upon the evaluated data [4]); then, we extended it with gene clustering and term-term similarity weights (SIM) to improve the correctness of predicted annotations [5] [6]. We implemented these methods within the *BioAnnotationPredictor* software component (Section 3), which is able to generate ranked lists of predicted annotations. It also saves the decomposed matrices generated by the employed technique, which can then be used for gene (or protein) semantic similarity computation. For this purpose, we developed *SimilBio* and *SemSim*, two novel software components.

We organize the rest of the paper as follows. After this Introduction, in Section 2 we discuss some related work and available tools about semantic similarity of genes. Then, in Section 3 we describe and discuss some main aspects of the *BioAnnotationPredictor* software, which we previously developed for the prediction of biomolecular annotations. It and its predicted annotations are leveraged in the novel *SimilBio* Web application, which we developed to evaluate the semantic similarity between genes, as described in Section 4. Programmatic access to the created functionalities for the computation of gene semantic similarity is provided by our newly developed Web service, called *SemSim*; we describe and discuss it, together with the Latent Semantic Indexing algorithm that it uses and some related use cases, in Section 5. Finally, we provide some interesting conclusions and envisage future developments in Section 6.

2 Related work

Computing semantic similarity between two genes (or proteins) is a key point to discover relationships between different genes (or proteins). Semantically similar genes (proteins) have many annotations in common and can have similar

- *D. Chicco and M. Masseroli are with the Dipartimento di Elettronica Informazione e Bioingegneria, Politecnico di Milano, Milan, Italy. D. Chicco is also with the Princess Margaret Cancer Centre, University of Toronto, Ontario, Canada.*
  E-mail: davide.chicco@gmail.com, marco.masseroli@polimi.it

functions, or be involved in similar pathways. Several semantic similarity measures, mainly based upon ontological annotations, exist. Some of them consider only the ontological topology of the controlled annotations upon which the similarity measure is based, while others also consider the corpus of available annotations. Numerous studies on semantic similarity between ontological terms (features) exist in literature: [7], [8], [9], [10]; some are regarding gene or protein similarities. To the best of our knowledge, the two most relevant tools that provide that functionality are *FastSemSim* [11] and *G-SESAME* [12].

FastSemSim [11] is an offline Python package which is able to compute the similarity between Gene Ontology feature terms and take advantage of this computation to calculate the semantic similarity between pairs of genes. The underlying idea of this approach is to first evaluate the functional similarity between all pairs of terms that annotate two genes and then combine these similarities in different ways to compute the overall gene similarity. *FastSemSim* lets the user choose among twenty similarity measures to use, all based upon the Gene Ontology tree structure [13].

G-SESAME [12] is a Web tool that first enables the user to upload a list of genes, then computes the functional similarity between them, and finally returns a clustered tree in which all genes with the same similarity value are located at the same tree height. The system uses measures such as the Resnik [7], Jiang [8] and Lin [9] ones to compute the similarity score between each pair of input genes; then it computes the weighted average of these scores.

The ontology structures of annotations may vary between ontology versions and many modifications may occur from one version to another. This is different from and more robust than both *FastSemSim* and *G-SESAME*. For the analysis of ontological annotations, we decided to use an information retrieval technique, which is independent from the data ontological structure. We chose Latent Semantic Indexing (LSI) [14], an algorithm able to take advantage of the relationships between GO terms and the genes annotated to them. LSI belongs to the category of the *vector-based approaches*, which work on a n-dimensional vector space where every considered entity identifies a dimension. We leveraged it for the computation of gene semantic similarities within two novel software components, named *SimilBio* and *SemSim*; the latter one, is a new Web service, here introduced for the first time, which we recently added to the online Bio Search Computing platform (*Bio-SeCo*) [15].

In contrast to *FastSemSim*, our semantic similarity functionalities are publicly available on the Internet, both through a programmatic and a friendly Web user interface, without the need of installing any additional software. Also *G-SESAME* is an online tool, but, as *FastSemSim*, it requires the user to input a specific list of genes to be compared other. Accordingly, the user has to know all the genes (and their IDs) for which he/she wants to compute the functional semantic similarity *a priori*. Conversely, both our online tools let the user input a single gene, compute the semantic similarity score between it and each available gene of the same organism and then return the list of all genes found functionally similar to the input gene, sorted by their similarity score to the input gene.

Thanks to its *SemSim* programmatic interface, another significant advantage of our software is the possibility to be easily composed with other Web services, in order to expand and enhance its results, as we did within the *Bio-SeCo* platform. In fact, results from any Web service registered within *Bio-SeCo* that is able to output gene IDs can be used as inputs to *SemSim*; similarly, *SemSim* outputs can be used as input data to other *Bio-SeCo* registered Web services that require gene IDs as input. Expansion of search results is one of the core features of Search Computing [16]; through these expansions, it supports users in finding answers to complex multi-topic biomedical queries by using a single online platform and leveraging a collection of available Web services. Furthermore, another important aspect of Search Computing is its ability to compose partially-ranked results and provide global-ranked expanded results [16]. We took advantage of this feature, since *SemSim* provides intrinsically ordered results, i.e. genes (by their similarity score).

In Table 1 we report the main differences between *SemSim*, *FastSemSim* and *G-SESAME*.

TABLE 1
Comparison of the main features of the evaluated tools. The “Web” column states if the tool is available online; “One vs. all” states if the tool allows comparing a single gene with all the available genes of the same organism; “One vs. some” states if the tool allows comparing a single gene with some other input genes; “Corpus indep.” states if the tool uses a semantic measure that is independent from the used data corpus and its ontology structure; “Rank. outp.” states if the output list of genes that the tool provides is ranked according to their similarity score; “Poss. exp.” states if the tool offers the possibility of expanding its output results by using them as input to other Web services or tools.

Tool	Web	One vs. all	One vs. some	Corpus indep.	Rank. outp.	Poss. exp.
SemSim	√	√		√	√	√
FastSemSim			√			
G-SESAME	√		√		√	

3 *BioAnnotationPredictor* SOFTWARE

BioAnnotationPredictor is a stand-alone software component that is able to efficiently run the SVD and SIM algorithms described in [5] in order to predict gene or protein functional annotations. It consists of two main modules: a C++ and a Java module, which interact through the Java Native Interface (JNI) programming framework. The C++ module is the algorithmic core of our software, which takes advantage of multiplatform and multithreading optimized mathematical libraries (i.e. AMD Core Math Library (ACML) [17] and SvdLibC [18]). The multithreading native part is developed using OpenMP (Open Multi-Processing) [19] compiler directives, which are leveraged by the algorithmic core independently from the operating system. The Java module manages the user interface, production of ontological graphs (using the Graphviz package [20]) and external data connections to the Genomic and Proteomic Data Warehouse (GPDW) [21], [22], [23], from which the available biomolecular annotations to be evaluated are extracted.

Besides producing ranked predicted annotations, the *BioAnnotationPredictor* also runs a pre-processing phase that generates serialized files of the gene matrix (U_k, in both the SVD and SIM methods), the singular value matrix (Σ_k) and the term matrix (V_k^T), respectively. The gene matrix file is then used by the *SimilBio* and *SemSim* software components to compute gene semantic similarities.

4 *SimilBio* WEB APPLICATION

SimilBio is a novel Web application developed using Java and Java Server Pages (JSP). It computes the semantic similarity between the genes of an organism, based upon their Gene Ontology annotations, and provides answers to user questions such as: *"Which are the human genes having semantic similarity level greater than 90% to the CHST14 human gene?"* *SimilBio* accomplishes this by running the LSI algorithm on the U_k gene matrix file generated by the *BioAnnotationPredictor* pre-processing phase, and providing the similar gene list through a Web interface for user browsing. It can also provide the list of Gene Ontology annotations predicted by the *BioAnnotationPredictor* for a gene identified by a user given Entrez Gene ID or gene symbol, together with the gene's additional details and GO annotations extracted from the GPDW. Additionally, it offers the same functionalities for proteins.

5 *SemSim* WEB SERVICE

SemSim is a new Representational State Transfer (REST) [24] Web service that we developed in JSP to provide programmatic access to the *SimilBio* functionalities. We registered and leveraged it within the Bio Search Computing system (*Bio-SeCo*): http://www.bioinformatics.deib.polimi.it/bio-seco/seco/ [15], which uses the Search Computing technology [16] to build answers to complex biomedical search queries. It does so by interacting with a collection of cooperating search services and using the ranking and joining of results as the dominant factors for service composition [25]. Within *Bio-SeCo*, the *SemSim* Web service can be queried individually, or together with the other Web services registered in *Bio-SeCo*, to answer complex multi-topic biomedical search questions such as: *"Which are the top ranked coexpressed human genes that are most significantly down regulated in tumor among the genes most functionally similar (i.e. with similarity greater than 95%) to the 'carbonic anhydrase IV' (CA4) gene and annotated to a known pathway?"*

5.1 Usage example

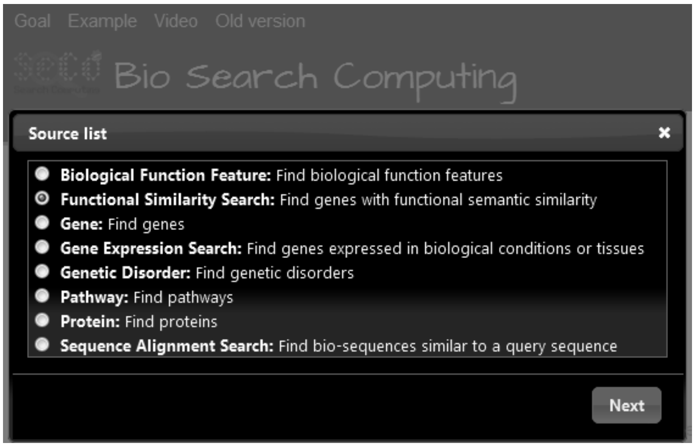
To take advantage of *SemSim*, a user has to access the *Bio-SeCo* website (http://www.bioinformatics.deib.polimi.it/bio-seco/seco/), start a new query session and then click on *Select Source*; this shows the list of topics addressed by the Web services registered within *Bio-SeCo* (Figure 1), which can be used to compose a search query. To use *SemSim*, the user has to select the second item of the list: *"Functional Similarity Search: Find genes with functional semantic similarity"*. Afterwards, the system asks the user to input the query parameters: the ID value (Gene ID) and source (Gene ID Name) of the input gene, the organism (Taxonomy ID) to which it belongs and the lower bound τ (Similarity Threshold) of the similarity level between the input gene and the compared genes of the same organism that the system should return. As a default example, the system proposes the Entrez Gene ID 368256 (which identifies the *paqr7b [progestin and adipoQ receptor family member VII, b]* gene) of the *Danio rerio* (zebrafish) organism and the $\tau = 0.7$ similarity threshold. By submitting these example values, the system returns the ordered list of known zebrafish genes most similar to the input gene, with a similarity score of at least 0.7, sorted by similarity level. This gene semantic similarity is computed through the LSI algorithm, which we describe next.

5.2 Latent Semantic Indexing (LSI)

Latent Semantic Indexing [14] is a Natural Language Processing (NLP) technique first proposed to discover and analyze relationships between sets of words and documents. Beyond NLP and document categorization, it has been widely used in many informatics-related fields such as search engine algorithms [26], e-commerce analysis techniques [27], image analyses [28] and programming code analyses [29]. In the bioinformatics context, LSI has been successfully used by Homayouni and colleagues in [30], where they applied it to the categorization of words in MEDLINE [31] paper abstracts. Its main principle is that words used in the same documents tend to have similar meaning. The LSI approach tries to find out the latent semantic structure given by the presence of particular words in certain documents, in order to categorize them and make them available for search queries.

In contrast to corpus-related measures (such as the Resnik [7], Jiang [8] and Lin [9] ones, all used by *FastSemSim* and *G-SESAME*), LSI is independent from the data used, and is not related to their ontological structure. This is why we decided to apply the LSI method to our genomic and proteomic scenario, where the structure of the ontologies used for biomolecular annotations varies often. Instead of words, we used genes, and instead of documents, we used biomolecular function features, described through the GO terms to which the genes are annotated.

After retrieving from GPDW [22] all the available genes of an organism and their Gene Ontology annotations, we first pre-process them through the *BioAnnotationPredictor* software component (see Section 3). It builds a matrix A, whose rows correspond to all genes of an organism



● **Biological Function Feature:** Find biological function features
○ **Functional Similarity Search:** Find genes with functional semantic similarity
● **Gene:** Find genes
● **Gene Expression Search:** Find genes expressed in biological conditions or tissues
● **Genetic Disorder:** Find genetic disorders
● **Pathway:** Find pathways
● **Protein:** Find proteins
● **Sequence Alignment Search:** Find bio-sequences similar to a query sequence

Fig. 1. Screenshot of the initial menu of the *Bio-SeCo* user interface. The list of topics covered by the services registered in *Bio-SeCo* for search computing is shown.

(identified by their gene ID value and source) and columns correspond to all their biomolecular function features (i.e. all the Gene Ontology feature terms associated with those genes). The $A_{i,j}$ element of the matrix is set to 1 if the gene i is annotated to the feature j, 0 otherwise. The resulting matrix is very sparse and large; so we use the truncated Singular Value Decomposition (tSVD) [2] to factor it and model the latent information as follow:

$$A \sim A_k = U_k \, \Sigma_k \, V_k^T \qquad (1)$$

where k is the truncation level of the SVD of A (Figure 2). Finally, the *BioAnnotationPredictor* generates and stores a serialized file for each of the U_k, Σ_k and V_k^T matrices.

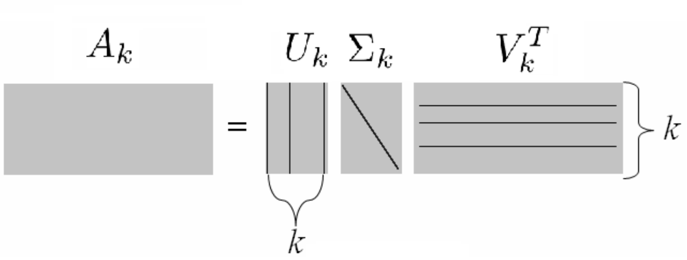


Fig. 2. Truncated Singular Value Decomposition (tSVD) matrices.

The orthogonal truncated matrices obtained by tSVD get precise names that are suggestive of their meaning:

- U_k: gene-vector matrix
- Σ_k: singular value matrix
- V_k^T: feature-vector matrix

These matrices can be used to measure the distances between objects (genes or features) in the k-dimensional space. For example, it is possible to compute the distance between two gene vectors to evaluate their similarity. This same computation can be done also for biological function features. In our implementation of the LSI, we chose to compute the Cosine similarity as the measure of semantic similarity between genes. The metric generally used for this vector similarity is:

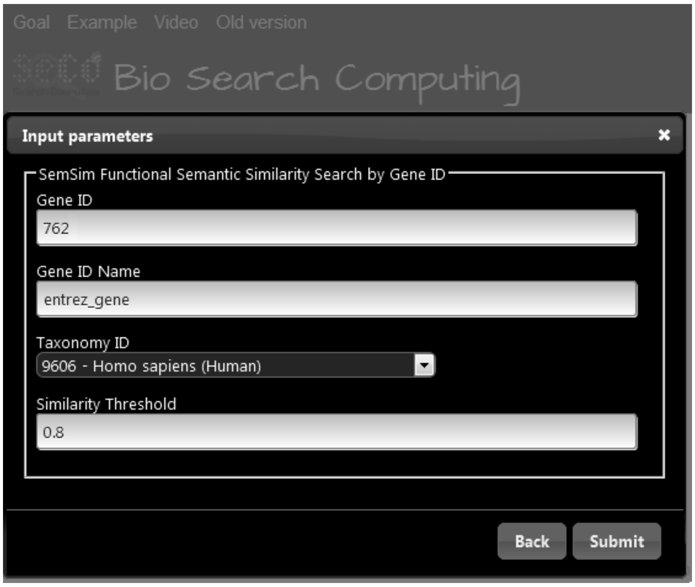
$$Cosine_{similarity}(v_1, v_2) = \frac{v_1 \cdot v_2}{||v_1|| \cdot ||v_2||} \qquad (2)$$

where v_1 and v_2 are the two vectors containing the gene feature sets, taken from the U_k gene vector matrix.

After receiving the user input gene ID (*Gene ID*), source (*Gene ID Name*) and organism (*Taxonomy ID*), and the threshold τ (*Similarity Threshold*), the implemented algorithm proceeds as follows:

1) retrieve the U_k serialized matrix that corresponds to the input organism
2) extract the g single row of the U_k matrix that corresponds to the input gene
3) for each u_i row of the U_k matrix different from g, compute: $score_i = Cosine_{similarity}(g, u_i)$
4) sort the $score$ vector in decreasing order
5) return all the u_i gene-vectors that have $score \geq \tau$

The output of the algorithm is a table in which each row corresponds to a gene and also each column corresponds to a gene, with rows ranked by their similarity score.



Input parameters

SemSim Functional Semantic Similarity Search by Gene ID

Gene ID: 762

Gene ID Name: entrez_gene

Taxonomy ID: 9606 - Homo sapiens (Human)

Similarity Threshold: 0.8

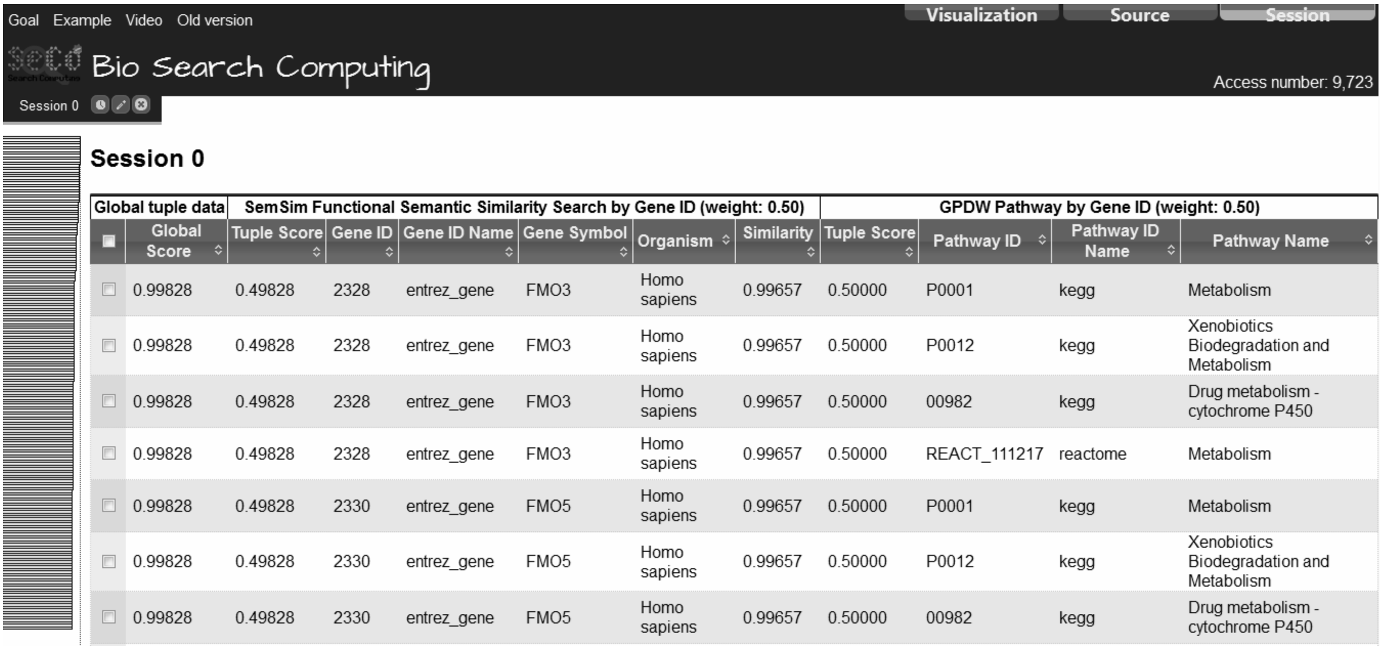
Fig. 3. User interface to set the input parameter values of the *SemSim - Functional Semantic Similarity Search by Gene ID* service registered in *Bio-SeCo*. Input values to search for human genes similar to the gene with Entrez Gene ID 762 (i.e. the *CA4 [carbonic anhydrase IV]* gene) are shown as an example.

5.3 Case studies

Here we report two example case studies that take advantage of the use of *SemSim* within the *Bio-SeCo* platform.

As first use case example, let us suppose that a scientist needs to explore available gene data to find the metabolic pathways (if they exist) in which the human genes most functionally similar (i.e. with similarity $score \geq 0.8$) to a given human gene X are involved. Using the resources registered in *Bio-SeCo* (Figure 1), the scientist can, for example, first run a functional similarity search by using our *SemSim* service as described in Section 5.1; he/she can do so to look for genes similar to an input gene X of the same organism (e.g. the gene with Entrez Gene ID *762* (http://www.ncbi.nlm.nih.gov/gene/?term=762[uid]), i.e. the human *CA4 [carbonic anhydrase IV]* gene). Figure 3 shows the *Bio-SeCo* interface where the user can specify and submit the input parameter values for such a search.

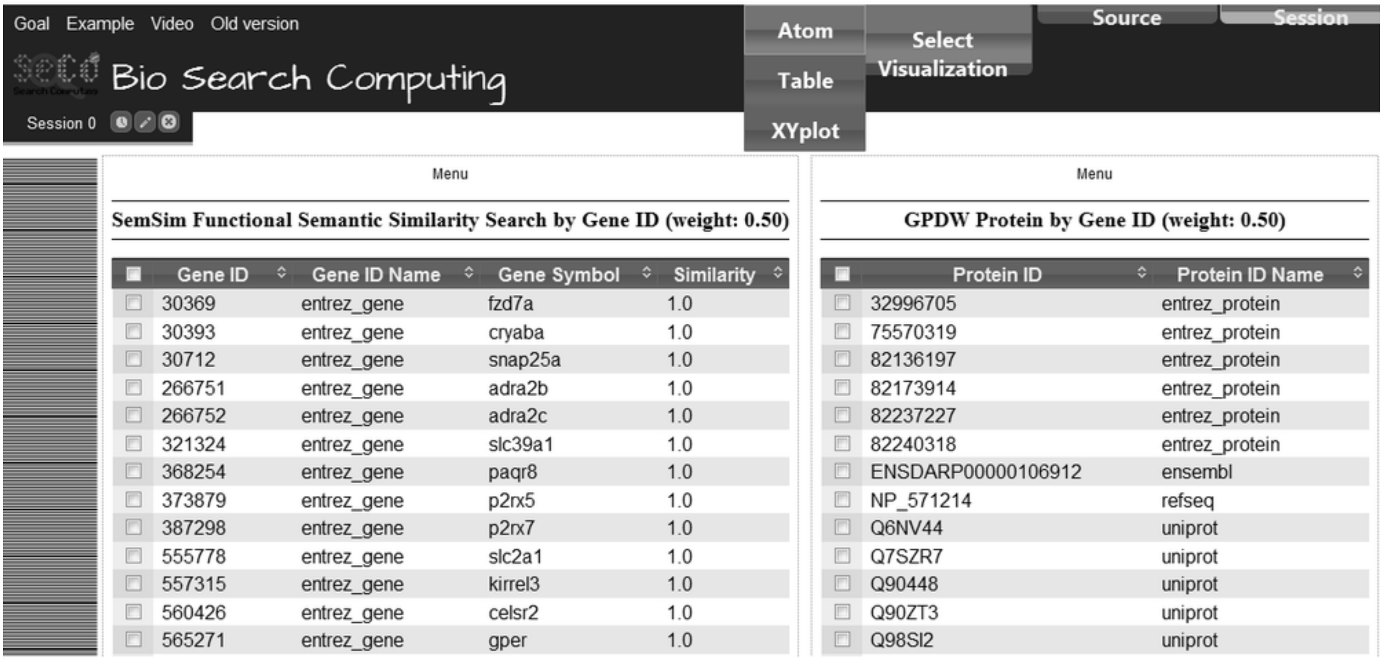
Obtained search results, including the details of the genes found to be most similar to the input gene, can be visualized in Atom or Table view. Then, the scientist can select all the most similar genes found, or only some of them (e.g. the ten most similar genes), and, automatically retrieve all the metabolic pathways associated with each of them; he/she can do so by using the *"GPDW - Pathway: Gene associated pathways by Gene ID"* query service, which is registered in *Bio-SeCo* as connected to the *SemSim* service. The top retrieved results are shown within the *Table* view in Figure 4; note that the left part of this table view contains the genes returned by the *SemSim* service, which were used as input for the *"Pathways by Gene"* query. Taking further advantage of the Search Computing principles implemented in *Bio-SeCo*, the user can further expand the obtained results, for example by checking if some of the genes found are known to be associated with a genetic disorder.



Session 0

Global tuple data		SemSim Functional Semantic Similarity Search by Gene ID (weight: 0.50)						GPDW Pathway by Gene ID (weight: 0.50)			
☐	Global Score	Tuple Score	Gene ID	Gene ID Name	Gene Symbol	Organism	Similarity	Tuple Score	Pathway ID	Pathway ID Name	Pathway Name
☐	0.99828	0.49828	2328	entrez_gene	FMO3	Homo sapiens	0.99657	0.50000	P0001	kegg	Metabolism
☐	0.99828	0.49828	2328	entrez_gene	FMO3	Homo sapiens	0.99657	0.50000	P0012	kegg	Xenobiotics Biodegradation and Metabolism
☐	0.99828	0.49828	2328	entrez_gene	FMO3	Homo sapiens	0.99657	0.50000	00982	kegg	Drug metabolism - cytochrome P450
☐	0.99828	0.49828	2328	entrez_gene	FMO3	Homo sapiens	0.99657	0.50000	REACT_111217	reactome	Metabolism
☐	0.99828	0.49828	2330	entrez_gene	FMO5	Homo sapiens	0.99657	0.50000	P0001	kegg	Metabolism
☐	0.99828	0.49828	2330	entrez_gene	FMO5	Homo sapiens	0.99657	0.50000	P0012	kegg	Xenobiotics Biodegradation and Metabolism
☐	0.99828	0.49828	2330	entrez_gene	FMO5	Homo sapiens	0.99657	0.50000	00982	kegg	Drug metabolism - cytochrome P450

Fig. 4. *Bio-SeCo* result Table View. The first seven results of the ”*Pathways by Gene*” query expansion for the genes most similar to the *CA4 [carbonic anhydrase IV]* (Entrez Gene ID: 762) gene are shown.



SemSim Functional Semantic Similarity Search by Gene ID (weight: 0.50)

☐	Gene ID	Gene ID Name	Gene Symbol	Similarity
☐	30369	entrez_gene	fzd7a	1.0
☐	30393	entrez_gene	cryaba	1.0
☐	30712	entrez_gene	snap25a	1.0
☐	266751	entrez_gene	adra2b	1.0
☐	266752	entrez_gene	adra2c	1.0
☐	321324	entrez_gene	slc39a1	1.0
☐	368254	entrez_gene	paqr8	1.0
☐	373879	entrez_gene	p2rx5	1.0
☐	387298	entrez_gene	p2rx7	1.0
☐	555778	entrez_gene	slc2a1	1.0
☐	557315	entrez_gene	kirrel3	1.0
☐	560426	entrez_gene	celsr2	1.0
☐	565271	entrez_gene	gper	1.0

GPDW Protein by Gene ID (weight: 0.50)

☐	Protein ID	Protein ID Name
☐	32996705	entrez_protein
☐	75570319	entrez_protein
☐	82136197	entrez_protein
☐	82173914	entrez_protein
☐	82237227	entrez_protein
☐	82240318	entrez_protein
☐	ENSDARP00000106912	ensembl
☐	NP_571214	refseq
☐	Q6NV44	uniprot
☐	Q7SZR7	uniprot
☐	Q90448	uniprot
☐	Q90ZT3	uniprot
☐	Q98SI2	uniprot

Fig. 5. *Bio-SeCo* result Atom View. The first thirteen results of the ”*Proteins by Gene: Get gene encoded proteins*” service for the genes most similar to the *Danio rerio* (zebrafish) gene *paqr7b [progestin and adipoQ receptor family member VII, b]* (http://www.ncbi.nlm.nih.gov/gene/?term=368256) are shown.

A second use case of interest concerns the answering the following multi-topic complex query: ”*Which are the proteins with the highest sequence similarity to the protein encoded by the genes in a given organism X that have the highest functional semantic similarity to a given gene Y?'*”

Once again, the user may start his/her query by using our novel ”*SemSim - Gene Functional Semantic Similarity Search by Gene ID*” service registered in *Bio-SeCo* (Figure 1 and Figure 3) to get the genes of the given organism X that are most functionally similar to a given input gene Y. He/she may then select, for example, the top twenty rows of the output table, i.e. the twenty most similar genes found, and use them as input to the ”*Proteins by Gene: Get gene encoded proteins*” service registered in *Bio-SeCo*. The user can then explore the obtained search results visualized in Atom view (e.g. see Figure 5), where he/she finds the details of the genes found most similar to gene X in the left table of the Atom view and the details of the proteins encoded by such genes in the Atom view right table. To complete the answer to the original multi-topic complex query, the user has to search for the protein that has the highest sequence similarity to the gene-encoded proteins found. Towards this aim, the user can take advantage of one of the ”*Sequence Alignment Search*” services in *Bio-SeCo* to further expand the obtained results. By selecting the ”*NCBI Blast: Protein sequence alignment search by Protein ID*” service and inserting all NCBI BLAST [32] parameters (as shown in Figure 3 of [15]), or using their default values, the user can run a final query that leads to a result table containing all proteins that satisfy the original query. Accordingly, the original multi-topic complex query is answered; then, the user may decide to further expand the final results by using another service in *Bio-SeCo*, or start a new query.

The possibility to easily construct in an explorative way complex biomedical queries, such as those of these case studies, and run them efficiently across multiple distributed sources permits the global evaluation of available bio-data; this can reveal unexpected results and lead to new discoveries of biomedical knowledge.

6 Conclusions

By using machine learning computational techniques, our software suite can predict reliable gene or protein functional annotations, as previously demonstrated. The new developed *SimilBio* Web application and *SemSim* Web service provide easy and fast public access to the predicted annotations, and allow them to be leveraged to compute semantic similarities between genes or proteins. In particular, the use of *SemSim* within our *Bio-SeCo* system allows for the support of the answering of complex biomedical questions and of biomedical knowledge discovery. In the future, we plan to integrate new computational biology Web services into *Bio-SeCo*, in order to provide new useful tools to scientists and researchers to address relevant biological problems.

Acknowledgments

The authors thank Marco Mina (Fondazione Bruno Kessler) for his help with the *FastSemSim* tool usage and Coby Viner (University of Toronto) who assisted with the proof-reading of this paper.

References

[1] G. Pandey, V. Kumar, and M. Steinbach, ”Computational approaches for protein function prediction: A survey”, *Twin Cities*: Department of Computer Science and Engineering, University of Minnesota, 2006.

[2] G. H. Golub, and C. Reinsch. ”Singular value decomposition and least squares solutions”, *Numerische Mathematik*, vol. 14.5, pp. 403–420, 1970.

[3] 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.

[4] D. Chicco, and M. Masseroli, ”A discrete optimization approach for SVD best truncation choice based on ROC curves”, *Proceedings of the 13th IEEE International Conference on Bioinformatics and Bioengineering (BIBE 2013)*, IEEE, pp. 1–4, 2013.

[5] D. Chicco, M. Tagliasacchi, and M. Masseroli, ”Genomic annotation prediction based on integrated information”, *Computational Intelligence Methods for Bioinformatics and Biostatistics*, Springer Berlin Heidelberg, pp. 238-252, 2012.

[6] M. Masseroli, M. Tagliasacchi, and D. Chicco. ”Semantically improved genome-wide prediction of Gene Ontology annotations”, *Proceedings of the 11th International Conference on Intelligent Systems Design and Applications - ISDA 2011*, pp. 1080-1085, 2011.

[7] P. Resnik, ”Using information content to evaluate semantic similarity in a taxonomy”, *Proceedings of the 14th International Joint Conference on Artificial Intelligence - IJCAI 95*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc, pp. 448–453, 1995.

[8] J. J. Jiang, D. W. Conrath, ”Semantic similarity based on corpus statistics and lexical taxonomy”, *Proceedings of the International Conference Research on Computational Linguistics - ROCLING X*, pp. 1–15, 1997.

[9] D. Lin, ”An information-theoretic definition of similarity”, *Proceedings of the 15th International Conference on Machine Learning*, Vol. 1. Citeseer, 1998, pp. 296–304.

[10] P. W. Lord, R. D. Stevens, A. Brass, and C. A. Goble, ”Investigating semantic similarity measures across the Gene Ontology: the relationship between sequence and annotation”, *BMC Bioinformatics*, vol. 19, no. 10, pp. 1275 – 1283, 2003.

[11] P. H. Guzzi, M. Mina, C. Guerra, and M. Cannataro, ”Semantic similarity analysis of protein data: assessment with biological features and issues”, *Briefings in bioinformatics*, vol. 13, no. 5, pp. 569–585, 2012.

[12] Z. Du, L. Li, C. F. Chen, S. Y. Philip, and J. Z. Wang, ”G-SESAME: web tools for GO-term-based gene similarity analysis and knowledge discovery”. *Nucleic acids research*, pp. W345–W349. 2009.

[13] P. H. Guzzi, and M. Mina, ”Towards the assessment of semantic similarity analysis of protein data: main approaches and issues”, *ACM SIGBioinformatics Record* vol. 2.3, pp. 17–18, 2012.

[14] S. Deerwester, ”Improving Information Retrieval with Latent Semantic Indexing”, *Proceedings of the 51st Annual Meeting of the American Society for Information Science*, vol. 25, pp. 36 – 40, 1988.

[15] M. Masseroli, M. Picozzi, G. Ghisalberti, and S. Ceri, ”Explorative search of distributed bio-data to answer complex biomedical questions”, *BMC Bioinformatics*, vol. 15, no. Suppl 1, S3, 2014.

[16] S. Ceri, D. Braga, F. Corcoglioniti, M. Grossniklaus, and S. Vadacca, ”Search computing challenges and directions”, Springer, Berlin Heidelberg, 2010.

[17] AMD Core Math Library (ACML), http://developer.amd.com/cpu/libraries/acml/

[18] D. Rohde, ”SVDLIBC”, http://tedlab.mit.edu/~dr/SVDLIBC

[19] L. Dagum, and R. Menon, ”OpenMP: an industry standard API for shared-memory programming”. *IEEE Computational Science & Engineering*, vol. 5, pp. 46–55, 1998.

[20] J. Ellson, E. Gansner, L. Koutsofios, S. C. North, and G. Woodhull, ”Graphviz - open source graph drawing tools”, *Graph Drawing*. Springer Berlin Heidelberg, pp. 483–484. 2002.

[21] A. Canakoglu, G. Ghisalberti, and M. Masseroli, ”Integration of biomolecular interaction data in a Genomic and Proteomic Data Warehouse to support biomedical knowledge discovery”, *Computational Intelligence Methods for Bioinformatics and Biostatistics*, Springer Berlin Heidelberg, pp. 112-126, 2012.

[22] A. Canakoglu, M. Masseroli, S. Ceri, L. Tettamanti, G. Ghisalberti, A. Campi, “Integrative warehousing of biomolecular information to support complex multi-topic queries for biomedical knowledge discovery”, *Proceedings of the 13th IEEE International Conference on Bioinformatics and Bioengineering (BIBE 2013)*, IEEE, pp. 1–4, 2013.

[23] F. Pessina, M. Masseroli, and A. Canakoglu, ”Visual composition of complex queries on an integrative Genomic and Proteomic Data Warehouse”. *Engineering* vol. 5, no. 10B, pp. 94–98, 2013.

[24] R. Fielding, ”Representational state transfer”, *Architectural Styles and the Design of Netowork-based Software Architecture*, pp. 76–85, 2000.

[25] D. Chicco, ”Integration of bioinformatics Web services through the Search Computing technology”, *Technical Report*, Dipartimento di Elettronica e Informazione, Politecnico di Milano, Milan, Italy, pp. 1–18, 2012.

[26] M. W. Berry, and M. Browne, ”Understanding search engines: mathematical modeling and text retrieval”, *SIAM*, Vol. 17, 2005.

[27] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, ”Analysis of recommendation algorithms for e-commerce”, *Proceedings of the 2nd ACM conference on Electronic commerce*, ACM, pp. 158–167, 2000.

[28] F. Monay, and D. Gatica-Perez. ”On image auto-annotation with latent space models”, *Proceedings of the eleventh ACM international conference on Multimedia*, ACM, pp. 275–278, 2003.

[29] J. I. Maletic., and A. Marcus, ”Using latent semantic analysis to identify similarities in source code to support program understanding”, *Proceedings of the 12th IEEE International Conference on Tools with Artificial Intelligence - ICTAI*, IEEE, pp. 46 – 53, 2000.

[30] R. Homayouni, K. Heinrich, L. Wei, M. W. and Berry, ”Gene clustering by latent semantic indexing of MEDLINE abstracts”. *Bioinformatics*, vol. 21, pp. 104-115, 2005.

[31] National Library of Medicine (US), “MEDLINE”, http://www.ncbi.nlm.nih.gov/pubmed

[32] M. Johnson, I. Zaretskaya, Y. Raytselis, Y. Merezhuk, S. McGinnis, T. L. Madden, “NCBI BLAST: A better web interface”. *Nucleic Acids Res*, vol. 36 (Web Server), pp. W5-W9, 2009.

Davide Chicco obtained his Bachelor of Science and Master of Science degrees in computer science at University of Genoa (Genoa, Italy) in 2007 and 2010. He then started the PhD program in computer engineering at Politecnico di Milano (Milan, Italy), where he graduated in Spring 2014. He has been a visiting research scholar at University of California Irvine (Irvine, California, USA), and since September 2014 he is a post-doctoral fellow at the Princess Margaret Cancer Centre, University of Toronto (Ontario, Canada). His research topics regard mainly machine learning algorithms applied to bioinformatics.

Marco Masseroli received the Laurea Degree in Electronic Engineering in 1990 from Politecnico di Milano, Italy, and a PhD in Biomedical Engineering in 1996, from Universidad de Granada, Spain. He is Assistant Professor at the Dipartimento di Elettronica, Informazione e Bioingegneria of Politecnico di Milano, and lecturer of Bioinformatics, Computational Biology and BioMedical Informatics. His research interests are in the area of bioinformatics, computational biology and biomedical informatics, focused on distributed Internet technologies, biomolecular databases, controlled biomedical terminologies and bio-ontologies to effectively retrieve, manage, analyze, and semantically integrate genomic information with patient clinical and high-throughout genomic data. He is the author of more than 150 scientific articles, which have appeared in international journals, books and conference proceedings.