EMBnet.news

Volume 14 Nr. 1
March 2008

- Browsing and Searching Gene Ontology Resources Using AmiGO
- GoPubMed: Answering biomedical questions
- The GOA@EBI resource

and more ...

Editorial

In this first issue of EMBnet.news in 2008 we are bringing to the attention of our readership a flavour of the diversity of our activity. You will find technical papers, course reports and a report on Bioinformatics activity in France. EMBnet adjusts itself to the needs of the community, and reveals its capability to play its role as a community of communities. On the year of its 20th anniversary, it is especially important that our activity is displayed. We would like to restress that we do accept contributions from any serious person working in Bioinformatics, not just EMBnet members. In this number you will find good examples of that. Also, we publish several featured articles (Ontology, GRID) that are part of collections of contributions that constitute real tutorials on these subjects. Again, we invite our readers to consider attending our conference in September and to actively contribute by submitting abstracts. The editorial board asks each and every reader to actively promote this publication. And please remember that you can give us feedback via the forum for EMBnet.news available at the EMBnet website.

The editorial board: Erik Bongcam-Rudloff, Domenica D'Elia, Pedro Fernandes, Kimmo Mattila and Lubos Klucar.

proteinspotlight
> ONE MONTH, ONE PROTEIN <

Protein Spotlight (ISSN 1424-4721) is a periodical electronic review from the SWISS-PROT group of the Swiss Institute of Bioinformatics (SIB). It is published on a monthly basis and consists of articles focused on particular proteins of interest. Each issue is available, free of charge, in HTML or PDF format at

`http://www.expasy.org/spotlight`

We provide the EMBnet community with a printed version of issue 91. Please let us know if you like this inclusion.

Cover picture: *Canna* (Italian Group) 'Yellow King Humbert'. Nairobi, 2008 [© Erik Bongcam-Rudloff]

Contents

Editorial Board:

EMBNET20YEARS

Domenica D'Elia
Conference Chair
Institute for Biomedical Technologies

Email:
domenica.delia@ba.itb.cnr.it
Phone: +39 080 5929674

EMBnet Conference 2008: 20th Anniversary Celebration

"Leading applications and technologies in Bioinformatics"

September 18-20, 2008 | Park Hotel San Michele

Martina Franca, Taranto, Italy

http://www.embnet.org/EMBnet20thAnniversary/

Overview

EMBnet (European Molecular Biology Network: http://www.embnet.org) is a worldwide network that was born in 1988 as association of bioinformatics service centers in Europe. Since 1988 EMBnet has been growing enormously by creating communities that share knowledge and resources by various means, extending nowadays to over 45 countries and reaching thousands of users.
This year EMBnet celebrate its 20th anniversary and to celebrate this special event we are organizing an international conference on the general theme:

Leading application and technologies in Bioinformatics

Our main goal is to present emerging trends that are likely to shape the future of Bioinformatics. We will have 4 half-day sessions on: 1) 'Omics', comparative studies and evolution; 2) Advanced bioinformatics technologies and applications; 3) Bioinformatics for biodiversity; 4) Training and e-Learning in Bioinformatics.

More detailed information is on the conference website.

This conference will provide an occasion of scientific exchange among Bioinformaticians and Life Sciences users from all over the world, including many of our old friends who contributed EMBnet's success in the past 20 years. Topics covered are as wide as possible mirroring the transversal scientific interests of the network.

“Leading applications and technologies in Bioinformatics”

We invite presentations on

- *Genomics*
- *Transcriptomics*
- *Proteomics*
- *Molecular structure prediction, modelling and dynamic*
- *System Biology*
- *Biobanks (databases and knowledgebases)*
- *Next generation sequencing: applications and case studies*
- *Text and data mining*
- *Biological data integration*
- *Ontologies*
- *Grid technologies and Web Services*
- *Molecular biodiversity and DNA barcode*
- *Metagenomics*
- *Training and e-Learning*

This conference will be to foster, as much as possible, new interactions and collaborations among the Bioinformatics community and Life Sciences scientists on common research goals. To achieve this aim, the Organising Committee has planned a few round table discussions on challenging research topics.

**The Organizing Committee cordially invites you to participate at this unique event
and enjoy the hospitality of Bari with the EMBnet community**

If you are interested, please submit a title and summary of your research for consideration of presentation.

To ensure the quality of the program, all proposals will be reviewed by the Scientific Committee.

Deadline for submission is May 15th, 2008

Information on abstract submission is available at the **conference website**.
Before submitting your abstract, please read the guidelines for authors.

For each abstract accepted, at least the present-ing author needs to register within the required time frame (June 30th, 2008).

For more information, please contact:

Domenica D’Elia
Conference Chair
Institute for Biomedical Technologies
http://www.itb.cnr.it/

Email: domenica.delia@ba.itb.cnr.it
Phone: +39 080 5929674

For exhibit and sponsorship information, please contact:

Francesca Mariani
Conference Organizing Secretariat
EEM Congresses & Events

Email: f.mariani@eemservices.com
Phone: 857-636-2332

Workshop Report

EMBRACE workshop on Applied Gene Ontology

Andreas Gisel

Institute for Biomedical Technologies, Bari, CNR, Bari, Italy

Figure 1: Introduction of the workshop by Prof. Cecilia Saccone, ITB Bari.

Gene Ontology (GO) is a well defined and structured vocabulary to describe gene products of all kinds of organisms on three levels: the function of the gene product, the biological process the gene product is involved in and its localization. In the last years the knowledge of GO increased drastically together with its use to analyse bioinformatic data. Nowadays there exists a large number of tools exploiting the GO knowledge to gain more information from bioinformatic and biological data able to provide a "biological picture" of the data set under investigation. Working more than four years using the GO and integrating information of the GO in bioinformatics analysis workflows it became clear that such a kind of information is very valuable but, beside by far not complete, it needs some explanation how to use and interpret it in the right way. Further, many scientists never heard about GO and to those who heard about GO it is not clear how to use it. Unfortunately, the GO is not a suitable resource for the biologist in the wet lab but more for the bioinformatician/analist who needs to develop solutions to analyze biological data. However, the biologist is the person who would mostly need this information. There is a wide range of bioinformatics tools using the knowledge of the GO to analyse, interpret, associate and sort biological data.

Within the Network of Excellence EMBRACE (`www.embracegrid.info`), the Institute for Biomedical Technologies (ITB) in Bari organized a workshop to disseminate and to give users insight into the GO and some GO tools such as browsers and tools for biological data annotation and microarray analysis to get a feeling on what the GO can provide to the end user. We included extensive hands-on sessions so that the users can test and better understand some of those tools. Another aim the workshop had was to initialize a discussion between the GO consortium, the tool developers and the end users to make each group conscious about the needs from each side, the strength and weaknesses of GO to achieve the best results by the GO knowledge use.

The workshop programme

The programme of this workshop was designed so that after the welcome and introduction note of Prof. Cecilia Saccone from ITB - Bari, we were presenting a general overview on the ontology theoretical concept, the gene ontology and the mechanism how the GO is linked with gene products of all kinds of species, to get the participants more familiar with the principles of the GO. After this theoretical section, we introduced the participants to different GO tools using the GO: GO browser, annotation tools, microarray analysis tools and others. Each tool was first presented on a theoretical level and consecutively on a practical level with an extensive hands-on session to give the participants a feeling of the power of the tools and how to use them. These hands-on sessions were only possible with the kind hospitality of the Department of Informatics of the University of Bari, represented by Prof. Floriana Esposito, Prof. Donato Malerba and Prof. Filippo Lanubile,

Figure 2: Midori Harris (EBI, Hinxton) introducing the concept of the gene ontology.

offering the use of one of their computer rooms for teaching and technical support.

The chosen tools were:

- the GO browser AMIGO (http://amigo.geneontology.org/cgi-bin/amigo/go.cgi);
- the tool GOpubMed for searching biological text using the GO (http://gopubmed.org/);
- the tool GOToolBox to analyze microarray results (http://crfb.univ-mrs.fr/GOToolBox/index.php);
- the tool Blast2GO for functional genomic research using BLAST and GO (http://bioinfo.cipf.es/blast2go/).

For details see the corresponding article within this issue of EMBnet.news.

In between we presented two projects in which we collaborate with local institutions such as our host, the Department of Informatics of University of Bari (Collaborative tagging for GO), and the National Institute of Nuclear Physics here in Bari (Gene Analogue Finder). Further, a short presentation about EMBnet and its main missions was made by Domenica D'Elia. Particular emphasis was put on services provided by EMBnet, such as the EMBnet e-Learning web portal, Quick Guides and the EMBnet.news, and about the interest of EMBnet to increase its collaborative activities.

The discussions were fruitful and fulfilled the workshop intention giving the participants a chance to understand better what the GO is and what it is good for, but also to initiate the interaction between end users, tool developers and the GO consortium. Another important result was that teachers were glad to contribute to this EMBnet.news issue with their articles on GO and GO tools presented at the workshop thus allowing to extend the benefit of this workshop to a wider community.

The programme, the presentations and a selection of pictures of the event are available on the workshop site http://beagle01.ba.itb.cnr.it/andreas/workshop/.

Programme:

Wednesday 7. November 2007

9:00 – 9:15 Introduction

9:15 – 9:45 Ontology: An Introduction, Claudia d'Amato Department of Informatics, University of Bari, Italy

9:45 – 10:15 Coffee

10:15 – 11:00 Gene Ontology, Midori Harris EBI, Hinxton, GB

11:00 – 11:45 GOA: Looking after GO annotations, Emily Dimmer EBI, Hinxton, GB (UniProt GOA project)

11:45 – 12:15 Insight into GO and GOA, Andreas Gisel ITB-CNR Bari, Italy

12:15 – 13:00 GO Browsers: AmiGO, Erika Feltrin CRIBI, University of Padova, Italy

13:15 – 15:00 Lunch

15:00 – 17:00 AmiGO hands-on with Coffee in parallel

20:00 Dinner

Thursday 8. November 2007

9:00 – 9:30 Gene Analogue Finder, Giulia De Sario ITB-CNR Bari, Italy and Angelica Tulipano INFN Bari, Italy

9:30 – 10:00 GO Annotation tools: GoPubMed, Andreas Doms Biotech TU Dresden, Germany

10:00 – 12:00 GoPubMed hands-on with Coffee in parallel

12:00 – 12:30 Collaborative tagging for GO, Domenico Gendarmi Department of Informatics, University of Bari, Italy

12:30 – 13:00 Discussion

13:00 – 14:30 Lunch

14:30 – 15:00 GO Microarray tools: GOToolBox, David Martin Genome bioinformatics lab, Center for Genomic Regulation, Barcelona, Spain

15:00 – 16:30 GOToolBox hands-on

18:30 Small Excursion and Dinner

Friday 9. November 2007

9:00 – 9:30 GO Other tools: Blast2GO, Ana Conesa Bioinformatics Department, Centro de Investigacion Principe Felipe, Valencia, Spain

9:30 – 11:30 Blast2GO hands-on with Coffee in parallel

11:30 – 13:00 Final Discussion

13:00 – 14:30 Lunch - End of the workshop

Acknowledgement:

The organization of the workshop would not have been possible without the financial support of the EMBRACE project (European Model for Bioinformatics Research and Community Education, Project Nr. 512092) and the help of Domenica D'Elia, Giulia De Sario, Cecilia Saccone, Angelica Tulipano, from the scientific and organizational point of view, and Serena Capodaqua, Lucrezia Cassano and Anita Tricarico for the administrative aspect.

Figure 3: Participants during the hands-on.

Course Report

Beijing, China, September 2007

Jingchu Luo

EMBnet China node, Center of Bioinformatics, Peking University, Beijing 100871, China

luojc@pku.edu.cn

It is a long tradition to give introductory courses for wet lab biologists since we joined EMBnet in 1997. The first course we organized was in April 1998 with five EMBnet teachers. More than dozen of EMBNet node managers came to teach during the past ten years. This article reports on the last 2007 course we organized: an Applied Bioinformatics course for 60 participants from the Institute of Botany, Chinese Academy of Sciences. Two EMBnet colleagues, Georgina Moulton and Josè R. Valverde visited Beijing to teach the course. Both the students and teachers were working very hard from 9:00am to 9:00pm every day. In addition to general introduction to various resources on the Internet, usage of Linux platform and EMBOSS command line tools, practical projects for the analysis of sequence, structure and function of bar-headed goose hemoglobin, carconoembryonnic antigen and spider toxin were also introduced.

Dates

1-6 September 2007

Students

60 graduate students (divided into 2 classes) from the Institute of Botany, Chinese Academy of Sciences (CAS)
Class 1 - 8:30am-8:30pm, 1, 3, 5 September
Class 2 - 8:30am-8:30pm, 2, 4, 6 September

Venue

EMBnet China node, Centre of Bioinformatics, Peking University, Beijing, China

Organisers

Jingchu Luo, Peking University, China
Song Ge, Deputy Director for Education, Institute of Botany, Chinese Academy of Sciences (CAS)

Teachers

Georgina Moulton, University of Manchester, UK, georgina.moulton@manchester.ac.uk
Jose R. Valverde, Centro Nacional de Biotechnologia, Spain, jrvalverde@cnb.uam.es
Jingchu Luo, Peking University, China, luojc@pku.edu.cn

Financial support

The international travel of GM was paid by the Institute of Botany, CAS, the local expenses was paid by Centre of Bioinformatics, Peking University. Both the international travel and the local expenses of Josè R. Valverde were paid by the Spanish EMBnet node.

Website

`http://abc.cbi.pku.edu.cn/`
`http://weblab.cbi.pku.edu.cn`

Environment

A training room with 35 PCs.
BioLand: a Linux bioinformatics environment with EMBOSS, Phylip, ClustalW, HMM, etc.
WinXP: with MEGA3.1, CN3D▯Swiss PDBViewer
WebLab: a web based platform developed locally

Participants of one class together with teachers Jose R. Valverde from Spanish node and Georgina Moulton from Manchester University.

Students working in the training room.

Lectures

Introduction to bioinformatics (JL)
Introduction to bioinformatics recourse (GM)
Introduction to Linux (JL)
Introduction to EMBOSS (GM)
Introduction to sequence analysis (GM)
Introduction to phylogenetic analysis (JL)
Introduction to protein family databases (GM)
Introduction to sequence motifs (GM)
Introduction to protein structure analysis (JL)

Hands-on sessions

Linux command
EMBOSS command line tools
WebLab protocols
Sequence similarity search (NCBI, EBI Blast)
Local Blast database search
Multiple sequence alignment and editing
Sequence motif search

Analysis of real problems using sequence and 3D analysis tools

Analysis of the high affinity of the hemoglobins of the bar-headed geese (the migratory birds flying over the Himalayas)
Analysis of the antigen-antibody interaction for the carcinoembryonic antigen
Analysis of the structural features of the spider toxin venom peptide
Analysis of the structure and function relationship of the metallothioneins

ReNaBi:
The French National Network of Bioinformatics platforms

Guy Perrière

Laboratoire de Biométrie et Biologie Évolutive, Université Claude Bernard – Lyon 1, France

Since December 2007, the French National Network of Bioinformatics platforms ReNaBi is the new French candidate node at EMBnet (`www.renabi.fr`), to be voted national node in 2008 AGM in Martina Franca (Italy). The ReNaBi is a structure that gathers 13 French bioinformatics centres from all over the country (Table 1). Each member platform is aimed at providing to the international community high-level technological resources in different bioinformatics fields. They follow the rules established by the Inter-Organisations Network (*Réseau Inter-Organismes* or RIO) national committee. Due to the fact that ReNaBi is a network of platforms, all the different fields of bioinformatics are covered in the services it provides: molecular evolution and phylogeny, genomics, transcriptomics and proteomics, structural biology, systems biology, etc.

ReNaBi aims

The ReNaBi aims to support the coordination of the activities of bioinformatics platforms and to optimize their scientific impact in life sciences. This is realized through different tasks:

- Providing access to biocomputing resources represented by a set of programs and publicly available data (such as general sequence databanks like EMBL, Ensembl or UniProt).

- Participating to methodological developments. The originality of ReNaBi is that the platforms are not only service providers but also integrate research teams in bioinformatics. Between 2006-07, the different ReNaBi research teams have published more than 300 papers in indexed journals.

- Development of specialized databases. As for the methodological developments, ReNaBi teams are also involved in the development and maintenance of recognized international databases (e.g., PRODOM, which is presently developed at the PRABI platform in Lyon).

The ReNaBi also contributes to scientific animation at a national level and support the promotion of the results completed by the teams participating to it. To achieve these goals, the members of ReNaBi coordinate their actions:

- When planning and organizing scientific animations, support and training.

- By supporting competences sharing and the re-use of programs developed on the different platforms.

- While joining to build projects involving several platforms, especially for the development of new tools.

- By communicating on the resources offered by the various platforms.

- While contributing to the initiatives aiming at structuring bioinformatics infrastructures on a European and international scale.

ReNaBi actions

The ReNaBi initiates actions at a national level. Those actions mainly consist in projects carried out in collaboration between several platforms and activities supported by a coordinating committee. National authorities (such as the French Ministry of Research or public research organisms) are solicited by ReNaBi to provide the means necessary to the realization of these actions. Any request for support for an action is under the responsibility of one of the coordinating committee member. The project handler must write a request for support (description of the project and justification of the support requested) that will be sent to the ReNaBi coordinator so that this one transmits it to the required national agencies. For each taken action the project handler will have to write a report in the month that will follow its

Name	Keywords	Place	URL
Plate-forme Bioinformatique de l'Institut Pasteur	Bactérial genomics, annotation and assembly	Paris	http://www.pasteur.fr/recherche/genopole/PF4/
Pôle Rhône-Alpes de Bioinformatique	Evolution, structural biology, databases, proteomics	Lyon and Grenoble	http://www.prabi.fr
Centre de Bioinformatique de Bordeaux	Yeast and functional genomics	Bordeaux	http://cbi.labri.fr/
Plateforme Bioinformatique de Jouy-en-Josas	Cartography, text mining	Jouy-en-Josas	http://migale.jouy.inra.fr/
Ressource Parisienne en Bioinformatique Structurale	Structural biology	Paris	http://bioserv.rpbs.jussieu.fr/
Plate-forme MicroScope	Genomes annotation	Evry	http://www.genoscope.cns.fr/
Unité de Recherche Génomique Info	Plant genomics	Versailles	http://urgi.versailles.inra.fr/
Plate-forme Bioinformatique de la Génopole Lille	Data analysis and data mining	Lille	http://www.genopole-lille.fr/
Information Génomique et Structurale	Functional genomics, bacterial annotation	Nice and Marseille	http://www.igs.cnrs-mrs.fr/
Plate-forme Bioinformatique de la Génopole Montpellier	Databases, compa-rative genomics, immunogenetics	Montpellier	http://www.genopole-montpellier-lr.org/PF/bioinfo/
Plate-formeBioinformatique de la Génopole Toulouse	Genetic mapping, expression data	Toulouse	http://bioinfo.genopole-toulouse.prd.fr/
Plate-forme Bioinformatique GenOuest	Databases, expres-sion data	Rennes, Nantes, Angers and Brest	http://genouest.org/
Plate-forme Bioinformatique de Strasbourg	Structural biology	Strasbourg	http://bips.u-strasbg.fr/

Table 1. List of the bioinformatics centres that are part of ReNaBi.

realization, and to forward it to the ReNaBi coordinator.

Structure and operation of the ReNaBi

The ReNaBi is animated by a coordinating committee, which is made of persons from each member platform and is chaired by one of them, designated as the coordinator (presently Antoine de Daruvar, from the CBiB platform in Bordeaux). The coordinating committee meets each year during a general assembly to which each platform of the network must send a representative. The coordinator sets up a steering committee made up of one or two persons (who are usually platform directors), which assist him/her in his/her missions. Among those missions are:

- To *represent* the network and to promote its activities with respect to national authorities or international.
- To *ask* for the means necessary to the operations of the network and the realization of its actions.
- To *transmit* to the national agencies the requests for support written by the project handlers.
- To *collect* the reports written by the persons in charge of the actions carried out and to produce an annual report of the network.
- To *organize* the annual general assembly of the coordinating committee.
- To *ensure* the organization, the animation and the writing of the report of the general assembly.
- To *consult* the coordinating committee on the important questions relating to the activity of the network.
- To *organize*, if necessary, additional meetings of the coordinating committee.

Moreover, each member of the coordinating committee can propose an action. In this case, he/she needs to:

- Writes the proposal that will be submitted to the national agencies to require a funding for the action requested.
- Writes and forwards to the coordinator a report in the month following the realization of the actions.

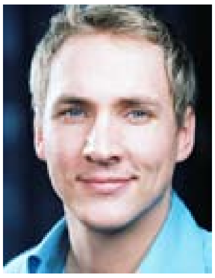
All the requests for actions are centralized by the coordinator who will monitor their diffusion within the coordinating committee before transmitting them to the national agencies. The requests for support that will be transmitted relate to two types of actions. First, there are collective actions, supported by the majority of the members of the coordinating committee. These actions will be transmitted in the name of the network. Second, there are actions corresponding to projects engaging only some platforms (at least two). For these actions, it is possible for any person in charge of a platform to contact the project handler if he/she wishes to be associated the project. The project handler will then decide freely to integrate the corresponding platform in the action.

Conditions of membership and withdrawal of the platforms

Is eligible to take part in ReNaBi, any French platform in bioinformatics having equipment and human means, and devoted to offer public services aimed at biologists. The platforms that wish to join the network must forward a request together with a presentation of the structure to the coordinator. An evaluation will be carried out by at least two members of the coordinating committee designated by the coordinator. The results of this evaluation (which will relate in particular to the level of opening to the public, the scientific impact, the quality and of the quantity of the services offered) will be presented at the coordinating committee and the decision will be made by a vote.

Each platform is free to withdraw from the network. For that, it is enough to inform the coordinator. At last, if a platform no longer seems to meet the criteria of eligibility, or if it does not respect the terms of the ReNaBi rules, the coordinating committee can decide of its exclusion. This will be achieved by a vote inwhich the reason for exclusion will be specified.

GoPubMed: Answering biomedical questions



Andreas Doms

Biotechnological Center, TU Dresden, Germany

This is a tutorial on how to use GoPubMed to answer biomedical questions. GoPubMed is a semantic search engine using background knowledge to help answering biomedical questions. It retrieves PubMed abstracts for your search query, then detects ontology terms from the Gene Ontology and Medical Subject Headings in the abstracts and finally allows the user to browse the search results by exploring the ontologies. A short introduction outlines the idea of ontology-based literature search, followed by an example on how to use GoPubMed. The tutorial can be followed online at www.gopubmed.org.

Introduction

When people search, they have questions in mind. Answering questions in a domain requires the knowledge of the terminology of that domain. Most search engines do not use background knowledge during the search. GoPubMed[2] allows to find answers by presenting search results in a structured way using the background knowledge of ontologies. The ontologies are used to categorize and explore literature abstracts. The current version of GoPubMed uses the Gene Ontology (GO) and the Medical Subject Headings (MeSH).

The Gene Ontology [1] was initially created with the goal to provide a structured, precisely defined, common and dynamic controlled vocabulary that describes the roles of genes and gene products in all organisms. The terms in GO are organized in three subontologies for cellular locations, molecular functions and biological processes. MeSH is a controlled vocabulary provided by the American National Library of Medicine and is used for indexing, cataloging, and searching for biomedical and health-related information and documents. It covers topics such as `Diseases`, `Organisms`, `Chemicals and Drugs`, `Techniques and Equipment` and others.

After a search GoPubMed shows the identified ontology categories on the left side as a tree, see also figure 1. The full branch of a category can be explored by expanding the respective nodes in the tree. Currently GoPubMed uses 36367 terms of the Gene Ontology and 31194 terms of MeSH.

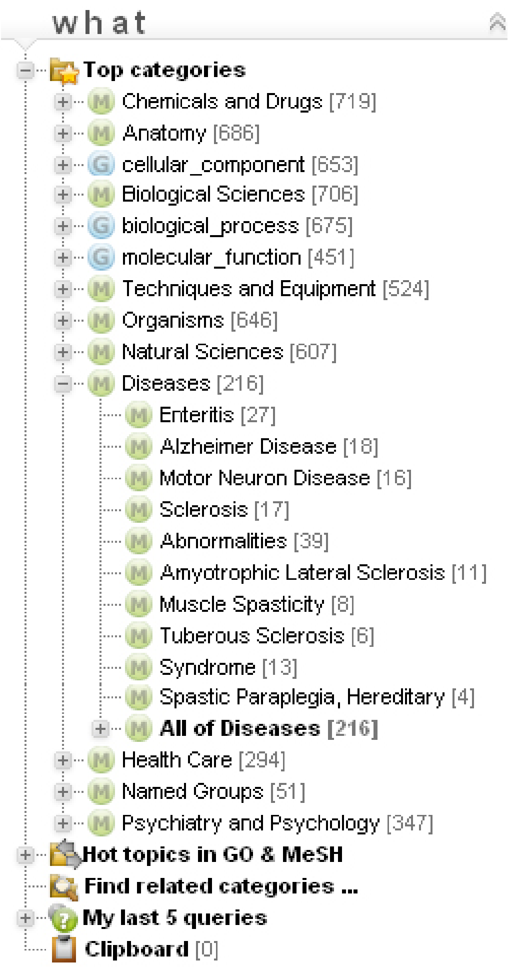
The query syntax of our search engine is the same as of PubMed. The retrieved results are exactly the same as one gets from PubMed. But in contrast to PubMed the abstracts are not presented as a long list. Abstracts and titles are analysed and searched for ontology terms. If a document contains an ontology term it is associated with the respective node in the ontology tree. The tree can be explored on the left side of the user interface. At any level a node can be selected by clicking on it. The documents on the right side are then condensed and will only show abstracts which mention the selected concept.

GoPubMed provides four answer sections: What, Who, Where and When.

The What section, see figure 1, holds the hierarchically structured abstracts of the initial PubMed query as described above. It also holds the "Hot Topics", a pre-computed bibliometric statistics on all ontology terms. Another option here is the personal clipboard. Just click on the clip next to each title to add this article to the clipboard.

The Who section, see figure 2, lists the authors with the most publications in the most recent publications within the search result. A name can represent several authors. Usually PubMed author names are highly ambiguous.

The Where section, see figure 3, list the journals mostly represented in the current search result. Other options here allow the user to filter the results for high impact journals or reviews.



what

- **Top categories**
  - M Chemicals and Drugs [719]
  - M Anatomy [686]
  - G cellular_component [653]
  - M Biological Sciences [706]
  - G biological_process [675]
  - G molecular_function [451]
  - M Techniques and Equipment [524]
  - M Organisms [646]
  - M Natural Sciences [607]
  - M Diseases [216]
    - M Enteritis [27]
    - M Alzheimer Disease [18]
    - M Motor Neuron Disease [16]
    - M Sclerosis [17]
    - M Abnormalities [39]
    - M Amyotrophic Lateral Sclerosis [11]
    - M Muscle Spasticity [8]
    - M Tuberous Sclerosis [6]
    - M Syndrome [13]
    - M Spastic Paraplegia, Hereditary [4]
    - M **All of Diseases [216]**
  - M Health Care [294]
  - M Named Groups [51]
  - M Psychiatry and Psychology [347]
- **Hot topics in GO & MeSH**
- **Find related categories ...**
- **My last 5 queries**
- **Clipboard** [0]

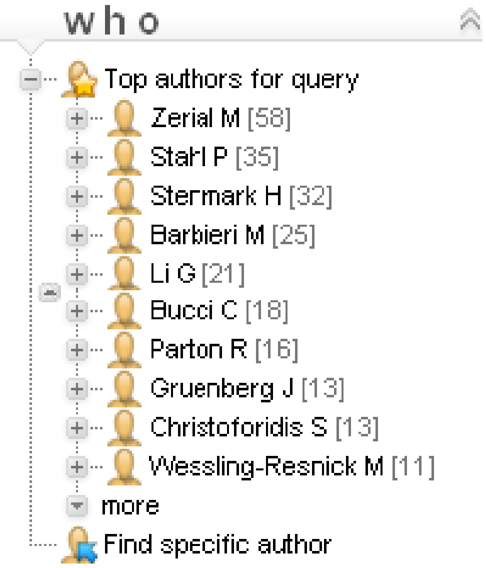
Figure 1. The What section shows hierarchically structured search results in GoPubMed.

The When section, see figure 4, lists the counts of publications of the recent years. An option allows to filter the search for the last week, month, year and others.

The four sections make answering biomedical questions easier. The advantage of GoPubMed's approach is that the sections hold answers which are directly connected to evidences in biomedical abstracts of PubMed.

Answering biomedical questions

GoPubMed can be used to answer biomedical questions. The given examples are not exhaustive but illustrate possible answers. Please follow the instructions here and explore GoPubMed at `www.gopubmed.org` while reading this tutorial.



who

Top authors for query
Zerial M [58]
Stahl P [35]
Stermark H [32]
Barbieri M [25]
Li G [21]
Bucci C [18]
Parton R [16]
Gruenberg J [13]
Christoforidis S [13]
Wessling-Resnick M [11]
more
Find specific author

Figure 2. The Who section shows the most active authors for a query.

The given evidences might change over time as new articles are being registered in PubMed.

What is the role of PrnP in mad cow disease?

1. Type "prnp" and click on the "Find it" button.

2. Prnp is a protein, so we use the protein name expansion. Click "Expand your query with synonyms for prnp".

3. We prefer high prole publications, so we choose "high impact journals only" in the Where. section.

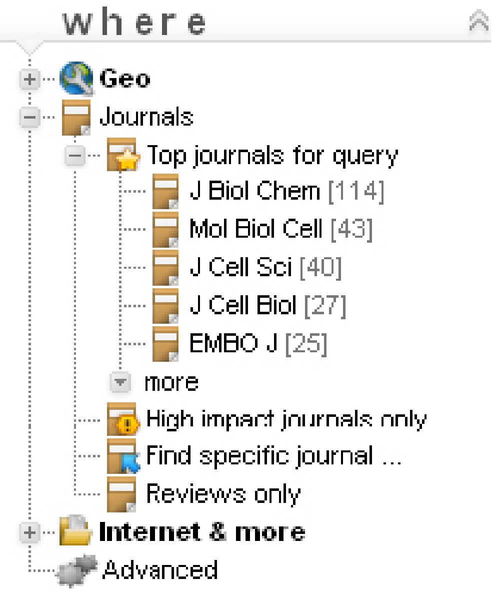
One answer we get is: *Prion diseases are caused by propagation of misfolded forms of the normal cellular prion protein PrP(C), such as PrP(BSE) in bovine spongiform encephalopathy (BSE) in cattle and PrP(CJD) in Creutzfeldt-Jakob disease (CJD) in humans. PMID: 17195841. The cited article is currently the very first in the list.*

What is the role of IDE in Alzheimer's disease?

1. Search for "IDE".

2. Expand the query with synonyms for the protein.

3. Click "high impact journals only".



where

Geo
Journals
Top journals for query
J Biol Chem [114]
Mol Biol Cell [43]
J Cell Sci [40]
J Cell Biol [27]
EMBO J [25]
more
High impact journals only
Find specific journal ...
Reviews only
Internet & more
Advanced

Figure 3. The Where section shows the most active journals for a query.

4. Browse the diseases branch.

Alzheimer Disease is the top term. Clicking on it shows a Science article from 2000 containing the following answer: *Recent studies suggest that insulin-degrading enzyme (IDE) in neurons and microglia degrades Abeta, the principal component of beta-amyloid and one of the neuropathological hallmarks of Alzheimer's disease (AD). (PMID: 11125142, Science).*

Which particular diseases are associated most often with HIV?

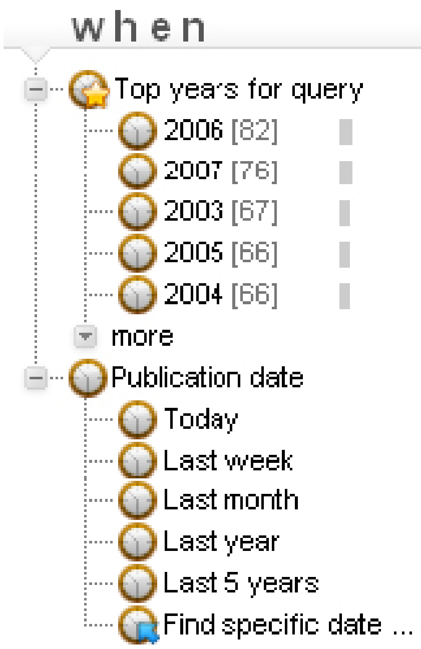
1. Search for "HIV".

2. In the What section browse to Diseases.

The most relevant topics in the category Diseases besides HIV Infections are Acquired Immunodeficiency Syndrome, Hepatitis, Tuberculosis, Viremia and Syphilis.

3. Click on the node Tuberculosis

This retrieves the relevant articles including statements such as "HIV and parasitic co-infections in tuberculosis patients".



when

Top years for query
- 2006 [82]
- 2007 [76]
- 2003 [67]
- 2005 [66]
- 2004 [66]
- more

Publication date
- Today
- Last week
- Last month
- Last year
- Last 5 years
- Find specific date ...

Figure 4. The When section shows the annual activity for a query.

Which techniques of treatment are used to help HIV patients?

4. Read the tooltip of Antiretroviral Therapy, which is the top category under Techniques and Equipment.

The definition is: *Antiretroviral Therapy are drug regimens, that aggressively suppress HIV replication. The regimens usually involve administration of three or more different drugs including a protease inhibitor.*

Who are the top authors for Antiretroviral Therapy and where was the research carried out and when?

5. Click on "Show statistics for term Antiretroviral Therapy, Highly Active".

The statistics show that B. Gazzard, J. Montaner and V. Soriano published most actively among others. Brian Gazzard published more than 100 articles from 1988 to 2007 with the Westminster Hospital, London, UK. Julio Montaner published more than 100 articles between 1988 and 2007 with Department of Medicine of University of British Columbia, Vancouver. Vicente Soriano published more than 80 articles in this field between 1991 and 2007 with the Servicio de Enfermedades Infecciosas, Instituto de Salud Carlos III, Madrid.

How does p53 affect apoptosis?

1. Search for "p53".

2. Filter for high impact journals.

3. Navigate to apoptosis in the branch of Biological Processes.

The title of this article suggests an answer to this question. *Transactivation of miR-34a by p53 Broadly Influences Gene Expression and Promotes Apoptosis. PMID: 17540599, Mol. Cell. , 2007.*

Who is a leading expert on liver transplantation in Germany?

1. Search for "liver transplantation Germany".

2. Expand the Who section.

P. Neuhaus is presented as a top author.

1. Click on the author's name.

This shows further information such as Berlin as his affiliation, his co-authors and the most relevant topics of his research. Peter Neuhaus is an internationally leading author in Liver Transplantation, Immunosuppressive Agents, Cyclosporine, Tacrolimus, Graft Survival and Graft Rejection. The author profile also lists another former affiliation and a link to his publications.

Where is the most research on celiac disease carried out?

1. Search for "celiac disease".

2. Expand the Where section.

3. Click on "Geo".

This reveals that Italy with Rome, Naples and Milan is most actively researching on this disease. Considering that the national dish in Italy is pasta, which contains a lot of gluten proteins, this might not surprise.

When was Paul Nurse publishing the most in his career?

1. Search for "Nurse P".

The displayed author profile states that the Nobel Prize winner is an internationally leading author in `Schizosaccharomyces`, `CDC2 Protein Kinase`, `Fugal Genes`, `Mitosis`, `Schizosaccharomyces pombe Proteins`, `Fungal Proteins` and `Cell Cycle Proteins`. He was awarded the Nobel Prize Physiology or Medicine in 2001 together with Leland H. Hartwell and R. Timothy Hunt for their discoveries regarding cell cycle regulation by cyclin and cyclin dependent kinases.

2. Expand the When section.

This shows that Paul Nurse was publishing 18 articles in 1991, his most active year. GoPubMed makes answering biomedical questions easier by structuring PubMed search results based on concepts identified in the titles and abstracts of articles using text-mining.

Community curation effort

Ontology-based literature search relies on sophisticated text-mining. While there are intelligent techniques to reach quality close to that of humans, those techniques depend on good training data. For an ontology-based search engine it is very important to distinguish the meaning of ontology terms in free text. This is in some cases a difficult task which needs training data for the machine learning algorithms.

GoPubMed offers a curation mode weaved into the web interface. Users can register with a single click, no login procedure is required. A browser cookie is saved and a new icon is shown.

From now on the curator can tag textmined concepts as "highly or less relevant for an article" as well as "incorrectly assigned" concepts with a single click. The curation data is anonymously used to evaluate and train the algorithms and thereby improve GoPubMed continuously.

The GoPubMed curation mode requires no login procedure. While registering as a curator the user is asked for his/her Email address. Later he will receive an email containing a link. With this link the user confirms his email address. The address is only used during the confirmation process and will not be published or given away in any case. Only curations made from confirmed addresses will be considered by the system later.

How to become a curator?

1. Click on this icon .

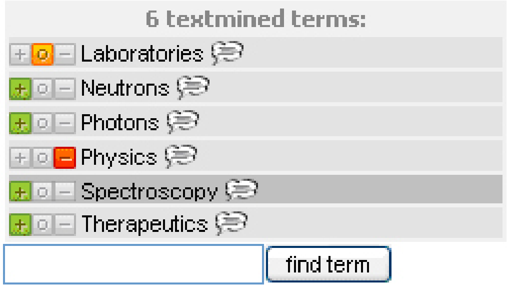
2. Read the explanation.

3. Enter your email address.

4. Search for your papers or for any other paper of an domain you are familiar with.

5. Click this icon: next to each article.

Next to the article appears a list of textminded terms, see figure 5.


6 textmined terms:

- ⊞ ⊡ ⊟ Laboratories
- ⊞ ⊡ ⊟ Neutrons
- ⊞ ⊡ ⊟ Photons
- ⊞ ⊡ ⊟ Physics
- ⊞ ⊡ ⊟ Spectroscopy
- ⊞ ⊡ ⊟ Therapeutics

Figure 5. Manually added curations by an author.

If you want to curate your own articles or other articles:

6. tag a concept as ⊞ `highly relevant` if it underlines the main focus of this research

7. tag a concept as ⊡ `less relevant` if it was correctly identified but not of importance for this research

8. tag general terminology as ⊟ `incorrect` if it has another meaning.

It is not necessary to curate all terms of an article. If you feel the abstract contains a concept which was not yet identified:

9. Click "Find term".

10. Enter the concept you expect.

11. If the appearing list contains it you may add it. Only concepts of GO and MeSH can be added.

The curations will be used to improve the search quality of GoPubMed. This process can be compared to tagging pictures on the web. It is much easier to classify pictures which were tagged by users. The anonymous curation data will be available for developers of other text-mining systems.

Author profiles

GoPubMed provides author profiles along with the literature search results. While the Who lists the authors with the most publications retrieved with the current search result. One author prole is always show above the documents view.

1. Search for "rab5", which is a protein involved in endocytosis.

An author prole of Marino Zerial is shown. Listed are his affiliations over the years, as well as the research topics he publishes the most about. A link is shown which reveals all publications known from the author. If we have the email address of the author an icon next to the authors name indicates that one can send a message to the author.

Author names in PubMed are highly ambiguous. Author disambiguation is a difficult task and the quality can vary from case to case depending on the ambiguity of author names. We counted over 3 million author names and computed 15 million profiles. The author disambiguation is based on article similarities. We use a bayesian approach as proposed in [3]. Users of GoPubMed can edit and improve the author profiles.

Summary

GoPubMed is a biomedical search engine using background knowledge to help answering questions. It introduces a new way of exploring search results by visualizing meta information on the search result in four answer sections on the left side. The What section structures the search result according to semantic concepts from two biomedical ontologies. The Who section lists prominent authors identified in the articles. The Where section holds information about the institutions and journals the current search results are related to. The When section groups articles by time periods.

GoPubMed provides a curation mode for identified ontology concepts. Users can give feedback about the quality of the identified concepts. This tool can be used to create large scale benchmarks in a collaborative way. The anonymous curation data are freely available.

Authors can update their profiles in GoPubMed. Author profiles contain information about the research topics of a person, his affiliations, his publications and his coauthors.

GoPubMed is freely available and is developed by the Biotechnology Center, TU Dresden in collaboration with the Transinsight GmbH.

Bibliography

[1] M Ashburner, CA Ball, JA Blake, D Botstein, H Butler, JM Cherry, AP Davis, K Dolinski, SS Dwight, JT Eppig, MA Harris, DP Hill, L Issel-Tarver, A Kasarskis, S Lewis, JC Matese, JE Richardson, M Ringwald, GM Rubin, and G Sherlock. Gene ontology: tool for the unication of biology. the gene ontology consortium. Nat Genet, 25(1):25{9, May 2000.

[2] A. Doms and M. Schroeder. Gopubmed: exploring pubmed with the gene ontology. Nucleic Acids Res, 33(Web Server issue), July 2005.

[3] Vetle I. Torvik, Marc Weeber, Don R. Swanson, and Neil R. Smalheiser. A probabilistic similarity metric for medline records: A model for author name disambiguation. JASIST, 56(2):140{158, 2005.

Browsing and Searching Gene Ontology Resources Using AmiGO

Midori A. Harris

EMBL—European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton, Cambs CB10 1SD, United Kingdom

Erika Feltrin

CRIBI, Università di Padova, via Ugo Bassi 58/B, Padova 35131 Italy

Introduction

At the EMBRACE Workshop on Applied Gene Ontology, we presented a hands-on tutorial focused on browsing and searching Gene Ontology (GO) data using the AmiGO browser. The tutorial followed three formal presentations that (1) provided an overview of the GO project; (2) described gene product annotation in some detail (see article by E. Dimmer in this issue); and (3) presented the key features of AmiGO.

The Gene Ontology (GO) project (`http://www.geneontology.org/`) is a collaborative effort to develop and use ontologies—structured, controlled vocabularies [1–4]—to support biologically meaningful, consistent and computable annotation of genes and their products in a wide variety of organisms. Participating groups include major model organism databases and other bioinformatics resource centres. The ontologies developed by the GO Consortium cover biological domains that are shared by all organisms. Three are attributes of gene products: molecular function describes activities at the molecular level, such as catalysis or binding; biological process describes biological goals accomplished by ordered assemblies of molecular functions; and cellular component describes locations, at the levels of subcellular structures and macromolecular complexes [5–14]. A fourth ontology, the Sequence Ontology (SO), covers sequence features [15,16].

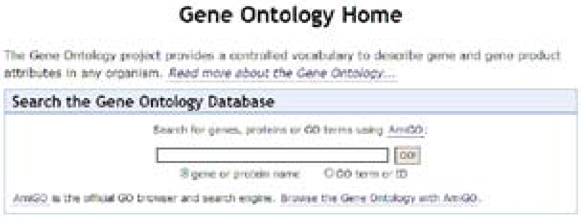
The ontologies in GO are structured as directed acyclic graphs (DAGs), which resemble hierarchies but allow any term to have one or more parents as well as zero, one, or more children. Within each vocabulary, terms are defined, and parent-child relationships are specified using is _ a and part _ of relationships (a third relationship type, "regulates", will be added early in 2008). GO ontology content and structure are described in more detail in online documentation and in references [5–9].

The annotation of gene products using GO terms is described online (`http://www.geneontology.org/GO.annotation.shtml`), in many publications (see the GO Bibliography, `http://www.geneontology.org/cgi-bin/biblio.cgi`) and in the article on GOA by E. Dimmer in this issue. AmiGO is an open-source web-based application designed to allows user to query, browse, and visualize data such as the GO vocabularies and gene product annotations. Although AmiGO can be used with any ontology available in Open Biomedical Ontologies (OBO) format [ref], it is best known as the "official" browser of the GO Consortium. At the EMBRACE workshop, the AmiGO tutorial was the first of several hands-on sessions that explored a variety of tools developed to use Gene Ontology terms and annotations in different contexts and for a range of purposes. The GO Consortium maintains a list of such tools on the web (`http://www.geneontology.org/GO.tools.shtml`).

AmiGO Tutorial

This tutorial aims to provide an introduction for biologists to the AmiGO browser (`http://amigo.geneontology.org`). It is organised based on common ways that a user may want to query GO: by GO term, by gene name or by protein sequence. There are also many other GO browsers, developed by outside groups, that can be used for this purpose (see `http://www.geneontology.org/GO.tools.browsers.shtml`).

The easiest way to use AmiGO is to go to the Gene Ontology web site (`http://www.geneon-`



Gene Ontology Home

The Gene Ontology project provides a controlled vocabulary to describe gene and gene product attributes in any organism. *Read more about the Gene Ontology...*

Search the Gene Ontology Database

Search for genes, proteins or GO terms using AmiGO:

[   ] GO!

● gene or protein name ○ GO term or ID

AmiGO is the official GO browser and search engine. Browse the Gene Ontology with AmiGO.

Figure 1: AmiGO simple search box is available at the Gene Ontology web site.

tology.org) where a simple search box is available (Figure 1).

As an alternative, the user can simply go to AmiGO home page and search the Gene Ontology database using the search interface (Figure 2). In the last year, AmiGO has been enhanced by the addition of new navigation and search options and an improved display of search results. At the moment, it provides an advanced search, a BLAST search and a browse option to easily navigate Gene Ontology vocabularies and annotations.

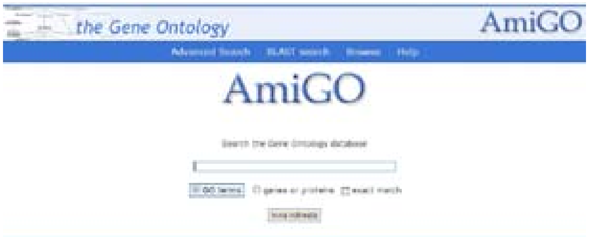
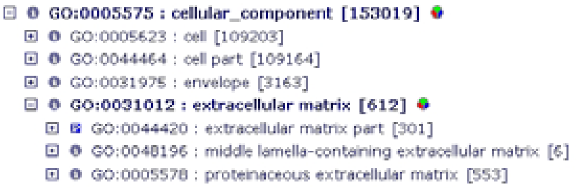


Figure 2: AmiGO home page at http://amigo.geneontology.org

Browse the Gene Ontology vocabularies

The Gene Ontology can be browsed using the "Browse" option. AmiGO displays GO terms as a tree organised into biological process, molecular function and cellular component branches (Figure 3).



⊟ ⓘ **GO:0005575 : cellular_component [153019]** ●
  ⊞ ⓘ GO:0005623 : cell [109203]
  ⊞ ⓘ GO:0044464 : cell part [109164]
  ⊞ ⓘ GO:0031975 : envelope [3163]
  ⊟ ⓘ **GO:0031012 : extracellular matrix [612]** ●
    ⊞ P GO:0044420 : extracellular matrix part [301]
    ⊞ ⓘ GO:0048196 : middle lamella-containing extracellular matrix [6]
    ⊞ ⓘ GO:0005578 : proteinaceous extracellular matrix [553]

Figure 3: A text representation of the ontology structure provided by AmiGO.

The first thing on each line can be either a ⊞, ⊟ or a ▪ icon. The ⊞ can be used to expand a node, showing all the children of the selected term. The ⊟ can be used to close the node, hiding the children. Finally, the ▪ means that the term on that line has no children. The next thing on each line can be either a P or ⓘ icon, which represent, respectively, a **part_of** or **is_a** relationship. Following each term is a number in parentheses. This shows the total number of gene products that have been annotated to this term or to a more specific term below this in the GO tree. The GO term identifier and term name can be clicked to get a more detailed view of the term, including the definition and all genes and gene products annotated to the term.

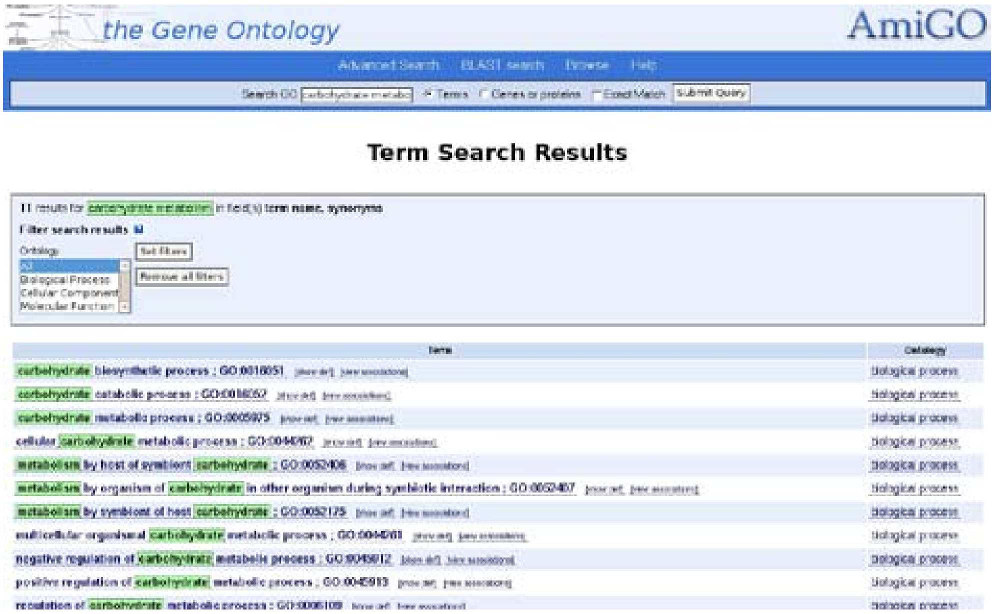
Terms may be followed by the ● icon. Clicking this icon will bring you to a pie chart which displays the percentage of gene products annotated to each term below that selected.

Following the term ID and name is a number in parentheses. This is the total number of genes manually annotated to this term and its children. Electronic annotations (evidence code IEA) are not shown for two reasons: there are large numbers of these annotations, and they are usually less specific than those that have been created or checked by a human.

Search by term name

It is also possible to search for term name (e.g. carbohydrate metabolism), and the result shows all GO terms that match the search string (Figure 4).

The results can be filtered to display only terms from one of the three GO vocabularies using the filtering options. Additionally, for any term in the list of results (e.g. regulation of catabolic process), the user can follow a link to a specific GO term detail page. The term detail page shows all the information available about the term: the term name and ID, any synonyms it might have, the term definition, its position in the GO structure, references to external databases, and the gene products associated with that term (Figure 5). In addition to a text representation of the ontology



Term Search Results

11 results for carbohydrate metabolism in field(s) term name, synonyms

Filter search results

Ontology: All, Biological Process, Cellular Component, Molecular Function

Term	Ontology
carbohydrate biosynthetic process ; GO:0016051 [show def] [view associations]	biological process
carbohydrate catabolic process ; GO:0016052 [show def] [view associations]	biological process
carbohydrate metabolic process ; GO:0005975 [show def] [view associations]	biological process
cellular carbohydrate metabolic process ; GO:0044262 [show def] [view associations]	biological process
metabolism by host of symbiont carbohydrate ; GO:0052406 [show def] [view associations]	biological process
metabolism by organism of carbohydrate in other organism during symbiotic interaction ; GO:0052407 [show def] [view associations]	biological process
metabolism by symbiont of host carbohydrate ; GO:0052175 [show def] [view associations]	biological process
multicellular organismal carbohydrate metabolic process ; GO:0044261 [show def] [view associations]	biological process
negative regulation of carbohydrate metabolic process ; GO:0045912 [show def] [view associations]	biological process
positive regulation of carbohydrate metabolic process ; GO:0045913 [show def] [view associations]	biological process
regulation of carbohydrate metabolic process ; GO:0006109 [show def] [view associations]	biological process

Figure 4. Display of results of a search for the string "carbohydrate metabolism".

structure, AmiGO provides a graphical view of the ontology.

To retrieve information about gene products annotated to the term itself or to the term and its children, the user can click on the number next the term. The genes or the gene products assigned to the term will then appear beneath the term information. The filtering menu can also be used to filter annotations by the database that supplied them, by the evidence code used in the annotation and by species.

From the results, the user can extract additional information about each individual annotated gene. The gene product detail page shows the information held in the GO database about that gene product, including all its GO annotations and the peptide sequence (if available).

The user can forward the FASTA protein sequence stored in the database to a BLAST search.

Search by gene name

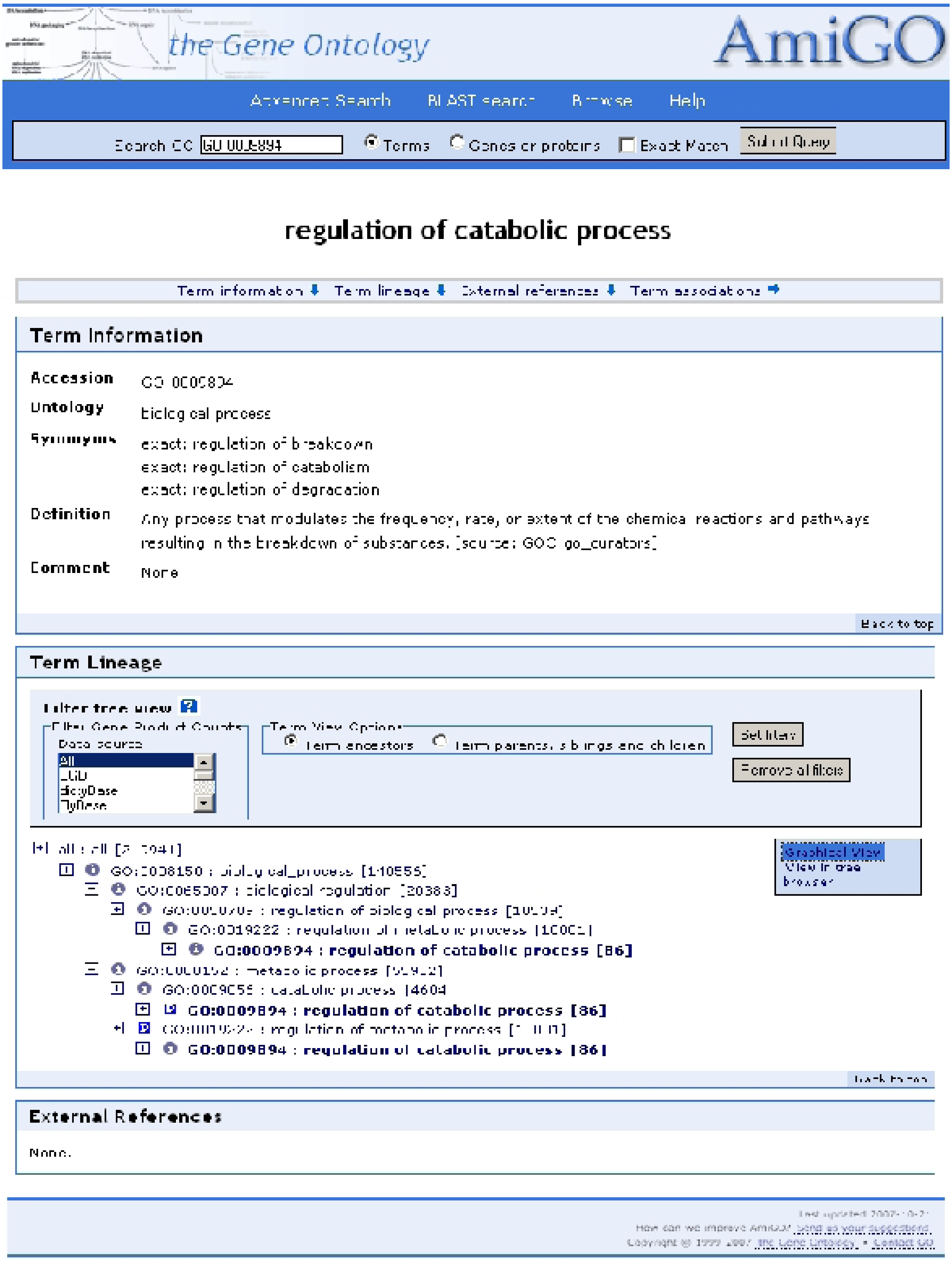
The Gene Ontology database can be searched also using a gene product name (e.g. grim). By default, the gene product search looks at gene product names, symbols and synonyms. To search by sequence accession or by database ID, an advanced search option is available. The results table lists the gene product(s) that match the query, with the part(s) of the gene product name that match the search term(s) highlighted in green. If the query match is in the gene product synonym(s), the matching synonym(s) will be displayed below the gene product name.

It must be remembered, however, that the Gene Ontology project is a work in progress and if a particular gene product is missing critical information, it is probably because it has not been annotated yet. A user can contact the GO via the GO Help mailing list (address below) to send suggestions for annotation.

BLAST searches in the GO database (GOst)

GOst is the Gene Ontology BLAST server, which allows you to perform a BLAST search on a protein sequence against all gene products that have a GO annotation. All of the groups that contribute GO annotations to the AmiGO database also contribute protein identifiers for their gene products. This is used to create a BLASTable database integrated with the GO annotations.

The AmiGO BLAST server searches the sequences from the GO protein sequence database, which comprises protein sequences submitted to the GO Consortium. Protein queries are searched



Search GO [GO:0009894] ● Terms ○ Genes or proteins □ Exact Match [Submit Query]

regulation of catabolic process

Term information ⬇ Term lineage ⬇ External references ⬇ Term associations ➡

Term Information

Accession GO:0009894

Ontology biological process

Synonyms exact: regulation of breakdown
exact: regulation of catabolism
exact: regulation of degradation

Definition Any process that modulates the frequency, rate, or extent of the chemical reactions and pathways resulting in the breakdown of substances. [source: GOC:go_curators]

Comment None

Back to top

Term Lineage

Filter tree view

Filter Gene Product Counts
Data source
All
CGD
dictyBase
FlyBase

Term View Options
● Term ancestors ○ Term parents, siblings and children

[Set filter]
[Remove all filters]

```
[+] all : all [[illegible]]
  [-] GO:0008150 : biological_process [110555]
    [-] GO:0065007 : biological regulation [20383]
      [-] GO:0050789 : regulation of biological process [[illegible]]
        [-] GO:0019222 : regulation of metabolic process [[illegible]]
          [+] GO:0009894 : regulation of catabolic process [86]
    [-] GO:0008152 : metabolic process [[illegible]]
      [-] GO:0009056 : catabolic process [4604]
        [+] GO:0009894 : regulation of catabolic process [86]
      [+] GO:0019222 : regulation of metabolic process [[illegible]]
        [-] GO:0009894 : regulation of catabolic process [86]
```

Graphical View
View in tree browser

Back to top

External References

None.

Last updated [illegible]
How can we improve AmiGO? Send us your suggestions.
Copyright © 1999-2007 the Gene Ontology • Contact GO

Figure 5: The term detail page for "regulation of catabolic process", showing all the information available about the term. The Graphical View option can be accessed by clicking on the hyperlink in the blue box to the right of the tree.

using BLASTP, while nucleotide sequences are searched using BLASTX.

There are three ways to submit a query sequence to the BLAST query: enter a UniProt accession ID (e.g. P55269); paste FASTA sequence(s) into the textbox; or upload a file containing sequences in FASTA format. GOst allows BLAST queries of up to 100 sequences but the total number of residues cannot exceed 3 million.

Conclusion

GO browsers like AmiGO allow scientists to find information on gene products involved in given processes, across a range of species. They remove the difficulties in searching that could be caused by ambiguous technical language and therefore open up the literature of unfamiliar fields for full investigation.

One of the continuing aims of the GO project is to encourage contributions from the scientific community, both to increase the number of annotations and to improve the quality of the GO vocabularies. Please send suggestions and comments to the GO help desk (use the link from AmiGO or email gohelp@geneontology.org). We appreciate your contributions, and will continue to improve the GO resources to help the biological research community work.

Acknowledgements

We thank Amelia Ireland and Judith Blake, who have presented tutorials upon which this tutorial is based. The Gene Ontology Consortium is supported by P41 grant HG002273 from the National Human Genome Research Institute (NHGRI), awarded to principal investigators Judith Blake, Michael Ashburner, Suzanna Lewis and J. Michael Cherry.

References

[1] Gruber, T. R. (1993). A translation approach to portable ontology specifications. Knowl. Acq., 5, 199–220.

[2] Jones, D. M. and Paton, R. (1999). Toward principles for the representation of hierarchical knowledge in formal ontologies. Data Knowl. Eng., 31, 99–113.

[3] Stevens, R., Goble, C. A. and Bechhofer, S. (2000). Ontology-based knowledge representation for bioinformatics. Brief. Bioinform., 1, 398–414.

[4] Schulze-Kremer, S. (2002). Ontologies for molecular biology and bioinformatics. In Silico Biol., 2, 179–193.

[5] The Gene Ontology Consortium (2000). Gene Ontology: tool for the unification of biology. Nat. Genet., 25, 25–29.

[6] The Gene Ontology Consortium (2001). Creating the Gene Ontology resource: design and implementation. Genome Res., 11, 1425–1433.

[7] Blake, J. A. and Harris, M. A. (2003). The Gene Ontology project: Structured vocabularies for molecular biology and their application to genome and expression analysis. In Baxevanis, A. D., Davison, D. B., Page, R. D. M., Petsko, G. A., Stein., L. D. and Stormo, G. (eds.) Current Protocols in Bioinformatics, John Wiley & Sons, New York.

[8] The Gene Ontology Consortium (2004). The Gene Ontology (GO) database and informatics resource. Nucleic Acids Res., 32, D258–D261.

[9] Harris, M. A., Lomax, J., Ireland, A. and Clark, J. I. (2005). The Gene Ontology project. In Subramaniam, S. (ed.) Encyclopedia of Genetics, Genomics, Proteomics and Bioinformatics, John Wiley & Sons, New York.

[10] Lewis, S. E. (2005). Gene Ontology: looking backwards and forwards. Genome Biol., 6, 103.

[11] Lomax, J. (2005). Get ready to GO! a biologist's guide to the Gene Ontology. Brief. Bioinform., 6, 298–304.

[12] Clark, J. I., Brooksbank, C. and Lomax, J. (2005). It's all GO for plant scientists. Plant Physiol., 138, 1268–1279.

[13] The Gene Ontology Consortium (2006). The Gene Ontology (GO) project in 2006. Nucleic Acids Res., 34, D322–D326.

[14] The Gene Ontology Consortium (2008). The Gene Ontology project in 2008. Nucleic Acids Res., 36, D440–D444.

[15] Eilbeck, K. and Lewis, S. E. (2004). Sequence Ontology annotation guide. Comp. Funct. Genomics, 5, 642–647.

[16] Eilbeck, K., Lewis, S. E., Mungall, C. J., Yandell, M., Stein, L., Durbin, R. and Ashburner, M. (2005). The Sequence Ontology: a tool for the unification of genome annotations. Genome Biol., 6, R44.

The GOA@EBI resource

Emily Dimmer

The Gene Ontology Annotation Group at the EBI

Emily Dimmer, Rachael Huntley, Daniel Barrell, David Binns, Claire O'Donovan, Rolf Apweiler

European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton, Cambridge, CB10 1SD, United Kingdom.

Introduction

At the EMBRACE workshop on Applied Gene Ontology, Emily Dimmer provided a presentation on the GOA@EBI resource that is run by the European Bioinformatics Institute.

As described in the accompanying report describing the AmiGO resource by Midori Harris and Erika Feltrin, the GO project is a collaborative effort that has created three ontologies of terms to describe the molecular functions, biological processes and subcellular locations that a gene product (from any species) might normally carry out or be involved in (examples of GO terms include: 'angiotensin receptor binding', 'ureteric bud branching' and 'voltage-gated calcium channel complex').

The GOA group is one of 15 member databases in the GO Consortium (GOC) (`http://www.geneontology.org`). GOA has been a member since 2001 and aims to provide high-quality assignments of Gene Ontology (GO) terms to proteins described in the UniProt (Swiss-Prot and TrEMBL) Knowledgebase (UniProtKB) (1). GOA now provides annotations to proteins from over 150,000 different taxonomic groups, and within the GOC the group is primarily responsible for the integration and release of GO annotations to the human, chicken and cow proteomes. The GOA groups creates both manual and electronic GO annotations, and by also integrating manual annotations created by 20 other curation groups, the GOA files have become a key dataset and a comprehensive source of GO annotation for all species (2).

This report will briefly outline the ways in which GO annotations are created and applied by researchers.

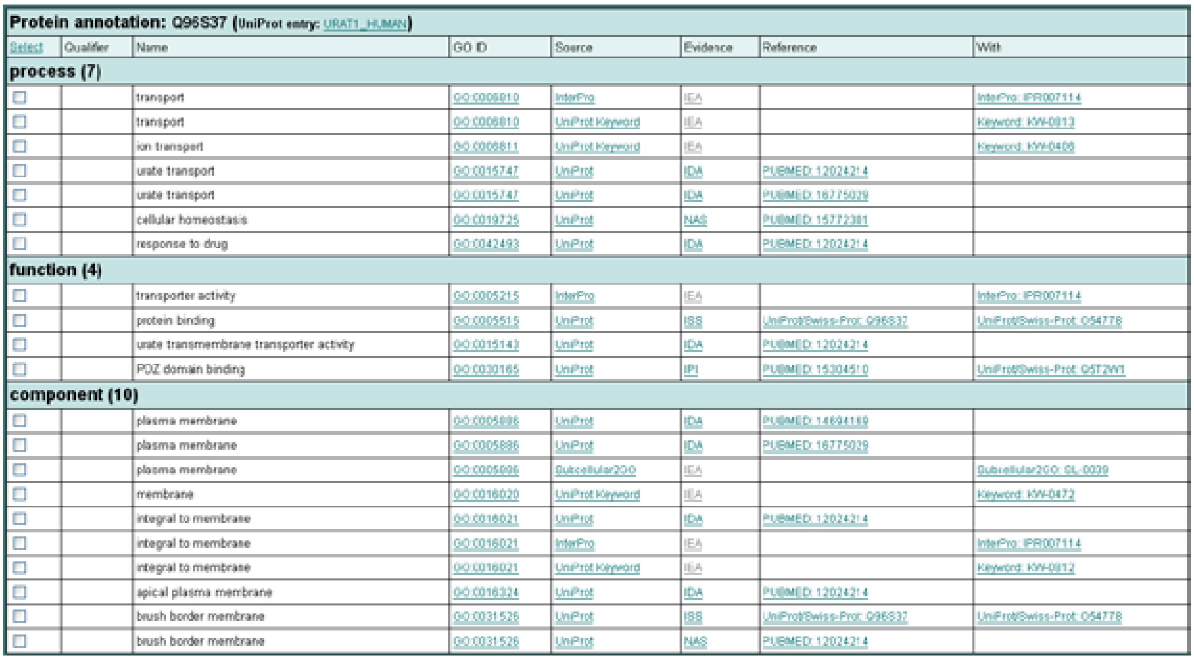
Uses of GO Annotation

GO annotations have proved to be remarkably useful for the mining of functional and biological significance from very large datasets, such as those resulting from microarray or proteomics experiments. Effective analysis of such results requires high-quality sources of standardised data presented in a configurable structure, as it becomes too difficult and time-consuming for researchers to comprehensively review current functional knowledge available for their entire set of gene or protein sequences. The vast majority of the popular functional analysis tools used by microarray and proteomics groups now use data supplied by the GOC. The standardized annotations are provided within an ontology structure that can be used to provide a broad overview of functional data for a range of biological functions or can help focus the investigator to a particular area of biology that is over-represented in the annotations provided for the proteins/genes in their dataset (3).

GO annotation data has been applied to help answer diverse questions; enabling investigators to validate experimental approaches (4), understand the underlying mechanisms observed during tissue or organ development (5), or the pathways affected in different multi-factorial disease states, such as schizophrenia, obesity and cancer (6,7,8) or transplant reactions (9). GO data has also applied to help select biomarkers for improved diagnostics and prognostics, identify novel therapeutics and evaluate effects of drug treatments (10, 11).

GO Annotation Methods

GO annotations are basically associations of specific GO terms to gene or protein identifiers (GOA applies UniProtKB protein accession numbers, e.g. Q9UBU3). Depending on the amount of functional data available, gene/protein identifiers can be annotated to multiple GO terms at any



Protein annotation: Q96S37 (UniProt entry: URAT1_HUMAN)

Select	Qualifier	Name	GO ID	Source	Evidence	Reference	With
process (7)							
☐		transport	GO:0006810	InterPro	IEA		InterPro: IPR007114
☐		transport	GO:0006810	UniProt Keyword	IEA		Keyword: KW-0813
☐		ion transport	GO:0006811	UniProt Keyword	IEA		Keyword: KW-0406
☐		urate transport	GO:0015747	UniProt	IDA	PUBMED: 12024214	
☐		urate transport	GO:0015747	UniProt	IDA	PUBMED: 16775029	
☐		cellular homeostasis	GO:0019725	UniProt	NAS	PUBMED: 15772381	
☐		response to drug	GO:0042493	UniProt	IDA	PUBMED: 12024214	
function (4)							
☐		transporter activity	GO:0005215	InterPro	IEA		InterPro: IPR007114
☐		protein binding	GO:0005515	UniProt	ISS	UniProt/Swiss-Prot: Q96S37	UniProt/Swiss-Prot: O54778
☐		urate transmembrane transporter activity	GO:0015143	UniProt	IDA	PUBMED: 12024214	
☐		PDZ domain binding	GO:0030165	UniProt	IPI	PUBMED: 15304510	UniProt/Swiss-Prot: Q5T2W1
component (10)							
☐		plasma membrane	GO:0005886	UniProt	IDA	PUBMED: 14694169	
☐		plasma membrane	GO:0005886	UniProt	IDA	PUBMED: 16775029	
☐		plasma membrane	GO:0005886	Subcellular2GO	IEA		Subcellular2GO: SL-0039
☐		membrane	GO:0016020	UniProt Keyword	IEA		Keyword: KW-0472
☐		integral to membrane	GO:0016021	UniProt	IDA	PUBMED: 12024214	
☐		integral to membrane	GO:0016021	InterPro	IEA		InterPro: IPR007114
☐		integral to membrane	GO:0016021	UniProt Keyword	IEA		Keyword: KW-0812
☐		apical plasma membrane	GO:0016324	UniProt	IDA	PUBMED: 12024214	
☐		brush border membrane	GO:0031526	UniProt	ISS	UniProt/Swiss-Prot: Q96S37	UniProt/Swiss-Prot: O54778
☐		brush border membrane	GO:0031526	UniProt	NAS	PUBMED: 12024214	

Figure 1. Annotation record in QuickGO (`http://www.ebi.ac.uk/ego`) for UniProtKB accession Q96S37, a human Urate anion exchanger. Both manual and electronic annotations are displayed, as well as PubMed references and evidence codes.

position in each of the three GO categories. In addition to a sequence identifier and a GO term identifier, annotations must also cite the source of evidence that supports a particular GO term-protein association and must state what type of evidence is provided by this source (represented by 13 different 'evidence codes'). GO annotations can be produced either by an annotator reading published scientific papers and manually creating each association or by applying computational techniques to predict associations. These two broad categories of techniques have their own advantages and disadvantages, but both require skilled biologists and software engineers to ensure that conservative, reliable annotation sets are produced. The annotations shown in Figure 1 display the range of electronic and manual annotations provided to a human anion exchange protein.

Electronic GO Annotation

The large-scale assignment of GO terms to proteins using computational methods is a fast and efficient way of associating high-level GO terms to a large number of genes, and with conservative usage these methods can produce reliable, although often less detailed annotations. And as the number of proteins requiring annotation increases exponentially with advances made in sequencing techniques, effective electronic annotation methods are becoming increasingly valuable. Out of the 150,000 taxonomic groups with GO annotations, over 99.9% only have electronic GO annotations assigned.

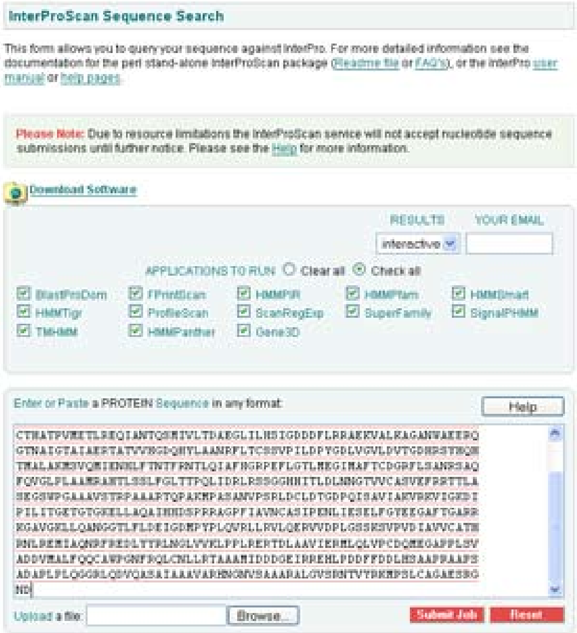
All computational annotations supplied by annotating groups are identified by the 'IEA' (Inferred from Electronic Annotation) evidence code and provide a reference that indicates the prediction method used (a full description of each individual electronic annotation method is displayed at: `http://www.geneontology.org/cgi-bin/references.cgi`). And although a range of computational techniques are being developed to predict functional attributes (12, 13, 14), there are currently two main electronic methods used by annotation groups in the GOC that provide sufficiently high-quality, conservative annotations. These two methods are the translation of external annotations into GO annotations (GO mappings) and the transfer of manual annotations to uncharacterised orthologs.

GO mappings are used to exploit external, well-established controlled vocabularies that have been used in external annotation efforts (examples of such vocabularies include Enzyme

Commission numbers or UniProt keywords). The GOC website provides 23 mapping files, each of which provide a 'translation' of external vocabularies to analogous GO terms (`http://www.geneontology.org/GO.indices.shtm`). GOA is responsible for the maintenance and development of two of these mappings; UniProt keywords (SPKW2GO) and UniProt subcellular location terms (SPSL2GO). For each set of mappings GO does not try to supersede the external system but to complement it. As the majority of such external concepts were developed for different purposes, the mappings can be neither complete nor exact.

Different GO mappings can result in the production of annotation sets of differing size and specificity, for instance the Enzyme Commission number to GO mapping file (EC2GO) produces a relatively low number of annotations (almost 600,000 annotations to a similar number of proteins in the GOA UniProtKB release 58), however the GO terms in the annotations applied using this mapping tend to be very specific and information-rich (e.g. EC:2.4.1.129 is mapped to 'peptidoglycan glycosyltransferase activity'; GO:0008955). In contrast, the InterPro2GO mapping provides a large number of annotations (over 17 million annotations to 3.4 million proteins), but to ensure annotations are correct, quite high-level, information-poor GO terms must sometimes be associated (e.g. InterPro:IPR000005 is mapped to 'intracellular'; GO:0005622). However an evaluation carried out on the UniProt keyword, E.C. and InterPro to GO mappings by the GOA group found that, when compared to existing manual annotations, the mappings predicted a correct GO term 91-100% of the time (14).

The **InterPro2GO mapping**, created and maintained by the InterPro group at the EBI, provides the highest electronic annotation coverage of all GO mappings (14, 16), supplying 41% of UniProtKB proteins with at least one electronic GO annotation (GOA UniProtKB release 58). InterPro integrates different protein signature recognition methods from the InterPro member databases (ProDom, PRINTS, SMART, TIGRFAMs, Pfam, PROSITE, PIRSF, SUPERFAMILY, Gene3D and PANTHER) and uses their combined methods to assign proteins to InterPro domains and families (15). InterPro curators create GO mappings by assigning a GO term to an InterPro identifier that correctly describes the function of all manually-annotated proteins in the UniProtKB/Swiss-Prot database that contain this domain, and then the same GO term is automatically applied to all UniProtKB proteins having the same domain.


InterProScan Sequence Search

This form allows you to query your sequence against InterPro. For more detailed information see the documentation for the perl stand-alone InterProScan package (Readme file or FAQ's), or the InterPro user manual or help pages.

Please Note: Due to resource limitations the InterProScan service will not accept nucleotide sequence submissions until further notice. Please see the Help for more information.

Download Software

RESULTS: interactive

YOUR EMAIL:

APPLICATIONS TO RUN ○ Clear all ● Check all

☑ BlastProDom ☑ FPrintScan ☑ HMMPIR ☑ HMMPfam ☑ HMMSmart
☑ HMMTigr ☑ ProfileScan ☑ ScanRegExp ☑ SuperFamily ☑ SignalPHMM
☑ TMHMM ☑ HMMPanther ☑ Gene3D

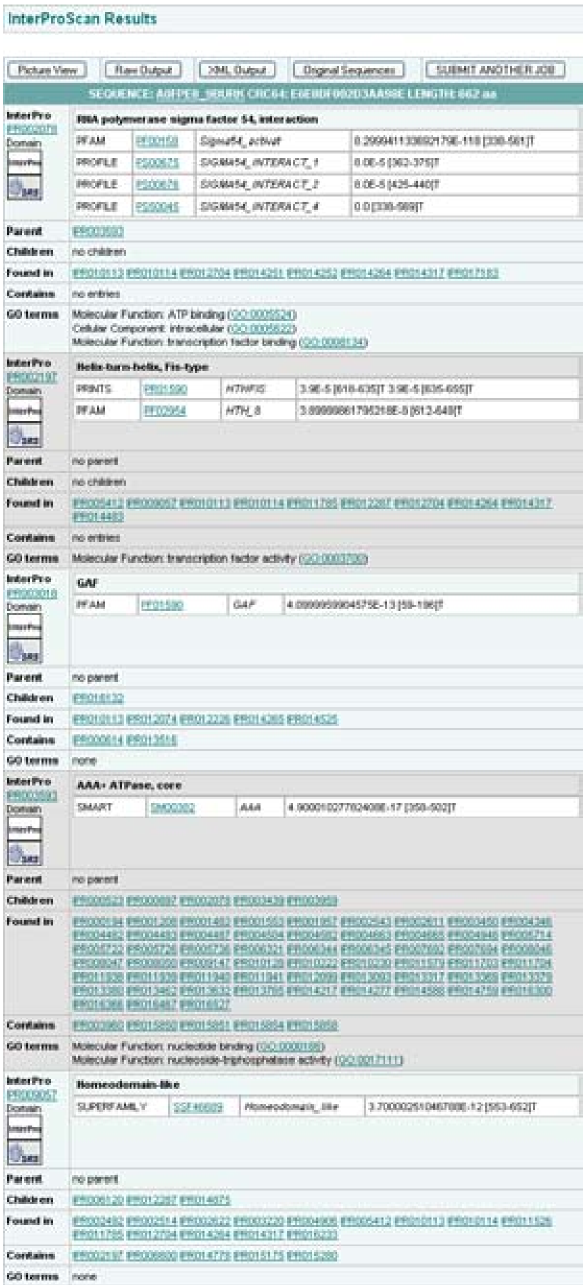
Enter or Paste a PROTEIN Sequence in any format:

```
CTHATPVRETLREQIANTQSRIVLTDAEGLILHSIGDDDFLRRAEKVALKAGANWAEERQ
GTNAIGTAIAERTATVVHGDQHYLAANRFLTCSSVPILDPYGDLVGVLDVTGDHRSTHQH
THALAKMSVQMIENKLFTNTFRNTLQIAFHGRPEFLGTLHEGIMAFTCDGRFLSANRSAQ
FQVGLPLAAMRANTLSSLFGLTTPQLIDRLRSSGGHNITLDLNNGTVVCASVEFRRTTLA
SEGSWPGAAAVSTRPAAARTQPAKMPASAMVPSRLDCLDTGDPQISAVIAKVRKVIGKDI
PILITGETGTGKELLAQAIHHDSPRRAGPFIAVNCASIPENLIESELFGYEEGAFTGARR
KGAVGKLLQANGGTLFLDEIGDMPYPLQVRLLRVLQERVVDPLGSSKSVPVDIAVVCATH
RNLREMIAQNRFREDLYYRLNGLVVKLPPLRERTDLAAVIERMLQLVPCDQMEGAPPLSV
ADDVMALFQQCAWPGNFRQLCNLLRTAAAMIDDDGEIRREHLPDDFFDDLHSAAPRAAPS
ADAPLPLQGGRLQDVQASAIAAAVARHNGNVSAAARALGVSRNTVYRKMPSLCAGAESRG
ND
```

Figure 2. The InterProScan submission page (http://www.ebi.ac.uk/InterProScan/).

The InterProScan service (16) (`http://www.ebi.ac.uk/InterProScan/`) applies the protein signature recognition methods from InterPro member databases to user-provided protein sequences (genomic sequences can also be queried in local installations of InterProScan). This service is free to all academic and commercial users and the web-based InterProScan service offers interactive or e-mail job submissions. Figure 2 shows the InterProScan job submission page, and Figure 3 the results produced from querying with the protein sequence for a bacterial transcription regulator. The results show six protein domains were recognised. Clicking on the 'Table View', 'Raw Output', or 'XML Output' buttons shows the GO annotation suggested by each match (Figure 3). InterProScan has proven particularly useful where groups need to quickly obtain a first round of GO annotations to new sequences.

Predicting GO terms based on sequence similarity. Another powerful electronic annotation



InterProScan Results

SEQUENCE: A0FPE8_9BURK CRC64: E6E8DF002D3AA98E LENGTH: 662 aa

InterPro IPR002078 Domain
RNA polymerase sigma factor 54, interaction

PFAM	PF00158	Sigma54_activat	0.299941133692179E-118 [330-501]T
PROFILE	PS00675	SIGMA54_INTERACT_1	0.0E-5 [362-375]T
PROFILE	PS00676	SIGMA54_INTERACT_2	0.0E-5 [425-440]T
PROFILE	PS50045	SIGMA54_INTERACT_4	0.0 [338-569]T

Parent: IPR003593
Children: no children
Found in: IPR010113 IPR010114 IPR012704 IPR014251 IPR014252 IPR014264 IPR014317 IPR017183
Contains: no entries
GO terms: Molecular Function: ATP binding (GO:0005524)
Cellular Component: intracellular (GO:0005622)
Molecular Function: transcription factor binding (GO:0008134)

InterPro IPR002197 Domain
Helix-turn-helix, Fis-type

PRINTS	PR01590	HTHFIS	3.9E-5 [618-635]T 3.9E-5 [635-655]T
PFAM	PF02954	HTH_8	3.89999861795218E-8 [612-648]T

Parent: no parent
Children: no children
Found in: IPR005412 IPR009057 IPR010113 IPR010114 IPR011785 IPR012287 IPR012704 IPR014264 IPR014317 IPR014483
Contains: no entries
GO terms: Molecular Function: transcription factor activity (GO:0003700)

InterPro IPR003018 Domain
GAF

PFAM	PF01590	GAF	4.09999959904575E-13 [59-196]T

Parent: no parent
Children: IPR016132
Found in: IPR010113 IPR012074 IPR012226 IPR014265 IPR014525
Contains: IPR000614 IPR013516
GO terms: none

InterPro IPR003593 Domain
AAA+ ATPase, core

SMART	SM00382	AAA	4.90001027702408E-17 [350-502]T

Parent: no parent
Children: IPR000523 IPR000897 IPR002078 IPR003439 IPR003959
Found in: IPR000134 IPR001208 IPR001482 IPR001553 IPR001957 IPR002543 IPR002611 IPR003450 IPR004346 IPR004482 IPR004483 IPR004487 IPR004504 IPR004582 IPR004663 IPR004665 IPR004846 IPR005714 IPR005722 IPR005726 IPR005736 IPR006321 IPR006344 IPR006345 IPR007692 IPR007694 IPR008046 IPR008047 IPR009050 IPR009147 IPR010128 IPR010222 IPR010230 IPR011579 IPR011703 IPR011704 IPR011938 IPR011939 IPR011940 IPR011941 IPR012099 IPR013093 IPR013317 IPR013369 IPR013379 IPR013380 IPR013462 IPR013632 IPR013765 IPR014217 IPR014277 IPR014588 IPR014759 IPR016300 IPR016366 IPR016487 IPR016527
Contains: IPR003960 IPR015850 IPR015851 IPR015854 IPR015856
GO terms: Molecular Function: nucleotide binding (GO:0000166)
Molecular Function: nucleoside-triphosphatase activity (GO:0017111)

InterPro IPR009057 Domain
Homeodomain-like

SUPERFAMILY	SSF46689	Homeodomain_like	3.70000251046708E-12 [553-652]T

Parent: no parent
Children: IPR006120 IPR012287 IPR014875
Found in: IPR002492 IPR002514 IPR002622 IPR003220 IPR004906 IPR005412 IPR010113 IPR010114 IPR011526 IPR011785 IPR012704 IPR014264 IPR014317 IPR016233
Contains: IPR002197 IPR006600 IPR014779 IPR015175 IPR015280
GO terms: none

Figure 3. InterProScan output showing predicted domains and GO term assignments (from InterPro2GO) for the protein sequence A0FPE8, a GAF modulated sigma54 specific transcriptional regulator in the bacterium B. phymatum.

method used to annotate an uncharacterised set of sequences is to transfer experimentally-verified manual annotations from well-characterised proteins to orthologs in a closely related species. Such annotations have resulted from a collaboration between the Ensembl and GOA groups. Orthology data from the Ensembl Compara method have been applied to transfer manual GO annotations between 1:1 and apparent 1:1 orthologs to 30 different species (`http://www.ebi.ac.uk/GOA/compara_go_annotations.html`), providing 34,025 annotations to 8,780 proteins (GOA UniProtKB release 58).

Manual GO annotation

As just mentioned, although computational methods can produce fast, large-scale assignments of GO terms to gene products, only few methods are used by the GOC and these heavily rely on existing manual annotation, and in many cases only achieve a high annotation coverage by applying information-poor GO terms (14). The manual annotation of genes using GO terms involves highly-trained curators reading published literature, evaluating the available experimental evidence and associating GO terms to a gene/protein record; in this way a detailed, information-rich summary of the activities, processes and subcellular locations known about a particular gene can be obtained. When much data has been published, the comprehensive annotation of a gene product may involve associating multiple, specific GO terms using evidence from many different publications. This work is labour-intensive and costly, however it yields a more detailed and accurate set of GO annotations than is possible from any computational approach. Information-rich, complete annotation sets are essential for users to be able to undertake a detailed functional analysis of their data.

Manual annotation methods also allow curators to describe the category of evidence that supported the association of a particular term to a protein, by selecting one of twelve evidence codes (this is in contrast to electronic annotations, which all use the same evidence code: 'IEA'). Such codes can indicate whether a function was inferred from experimental data, for instance from the results of a direct experimental assay: 'IDA' (e.g. from enzyme activity or cell fractionation assays), based on reviewed computational analysis, for instance from sequence or structural similarity 'ISS', or finally whether the annotations were based on an author's statement or curator judgement (see: `http://www.geneontology.org/GO.evidence.shtml`).

Manual annotations can also include 'qualifiers', which can have three values: 'co-localizes with' (to indicate a transient or peripheral association of the protein with an organelle or complex), 'contributes _ to' (where a function of a protein complex is facilitated, but not directly carried out by one of its subunits) and 'NOT' (to indicate situations where either authors have published conflicting data, or where in contrast to previous assumptions, a protein is not found to have a particular activity, location or process involvement). The 'NOT' qualifier produces the most drastic change in the interpretation of an annotation and users of large datasets are often advised to remove such qualified annotations before carrying out any large-scale functional analyses.

Manual annotation efforts

As manual annotation is a time-consuming and expensive activity for any database to undertake, many groups have to focus their efforts and have annotation priorities that reflect their user communities requirements. The GOA group is currently involved in collaborations which aim to focus on improving areas of GO annotation for human proteins involved in immunological- and cardiovascular-related processes (17). Both of these annotation efforts involve curators working closely with external experts to ensure that the annotation data provided is as complete and detailed as possible (for further information, see: (`http://wiki.geneontology.org/index.php/`Cardiovascular and `http://wiki.geneontology.org/index.php/Immunology`).

In addition, in 2006 GOA became a central participant in an NIH-funded 'Reference Genome' annotation project, which involves 12 diverse model organism groups from the GOC who are committed to the comprehensive annotation of conserved ortholog sets. With this project the GOC intends to generate a reliable set of GO annotations that will aid comparative methods used in first pass annotation of other proteomes (further information available at: `http://www.geneontology.org/GO.refgenome.shtml`).

Access to Annotation Datasets

Entire sets of GOA annotation files can be downloaded from both the GO (`ftp://ftp.geneontology.org/pub/go/gene-associations`) or GOA ftp sites (`ftp://ftp.ebi.ac.uk/pub/databases/GO/goa/`), where all annotation data are provided in simple 15-column formats, called a 'gene association file'. GOA provides one large gene association file for all annotations provided to UniProtKB protein accessions (gene _ association.goa _ uniprot), as well as individual files for a number of species, including: human, mouse, rat, Arabidopsis, zebrafish, chicken and cow.

GO annotation data can also be browsed using either the official GOC web browser AmiGO (`http://amigo.geneontology.org/cgi-bin/amigo/go.cgi`) or GOA's web browser QuickGO (`http://www.ebi.ac.uk/ego`). GO annotations are additionally imported into Ensembl and NCBI's Entrez Gene, as well as 100's of third party functional analysis tools.

Getting in Contact

If you have any queries regarding the GOA resources, please contact us at: `goa@ebi.ac.uk`. Alternatively, any questions or suggestions regarding the GO can be emailed to: `gohelp@geneontology`. If you would like to be kept informed of developments in the GO project, you can also join the 'go friends' e-mail list by sending a message to: `gofriends-request@geneontology.org` with the word 'subscribe' added to the body of the e-mail.

Funding

The GOA groups is supported by funding from a British Heart Foundation grant (SP/07/007/23671), a BBSRC Tools and Resources grant (BB/E023541/1), an NHGRI grant (HG002273), as well as core EMBL funding.

References

1. Camon E, Barrell D, Lee V, Dimmer E, Apweiler R. The Gene Ontology Annotation (GOA) Database--an integrated resource of GO annotations to the UniProt Knowledgebase. In Silico Biol. 2004;4(1):5-6.
2. Lee V, Camon E, Dimmer E, Barrell D, Apweiler R. Who tangos with GOA?-Use of Gene Ontology Annotation (GOA) for biological interpretation of '-omics' data and for validation of automatic annotation tools. In Silico Biol. 2005;5(1):5-8.

3. Khatri P, Draghici S. Ontological analysis of gene expression data: current tools, limitations, and open problems.Bioinformatics. 2005 Sep 15;21(18):3587-95.

4. Kislinger T, Rahman K, Radulovic D, Cox B, Rossant J, Emili A. PRISM, a generic large scale proteomic investigation strategy for mammals. Mol Cell Proteomics. 2003 Feb;2(2):96-106.

5. Campanaro S, Picelli S, Torregrossa R, Colluto L, Ceol M, Del Prete D, D'Angelo A, Valle G, Anglani F. Genes involved in TGF beta1-driven epithelial-mesenchymal transition of renal epithelial cells are topologically related in the human interactome map. BMC Genomics 2007; 8: 383-398.

6. Prabakaran, S., Swatton, J. E., Ryan, M. M., Huffaker, S. J., Huang, J. T., Griffin, J. L., Wayland, M., Freeman, T., Dudbridge, F., Lilley, K. S., Karp, N. A., Hester, S., Tkachev, D., Mimmack, M. L., Yolken, R. H., Webster, M. J., Torrey, E. F. and Bahn, S. (2004). Mitochondrial dysfunction in schizophrenia: evidence for compromised brain metabolism and oxidative stress. Mol. Psychiatry 9, 684-697.

7. Philip-Couderc, P., Pathak, A., Smih, F., Dambrin, C., Harmancey, R., Buys, S., Galinier, M., Massabuau, P., Roncalli, J., Senard, J. M. and Rouet P. (2004). Uncomplicated human obesity is associated with a specific cardiac transcriptome: involvement of the Wnt pathway. FASEB J. 18, 1539-1540.

8. Shi T, Liou LS, Sadhukhan P, Duan ZH, Novick AC, Hissong JG, Almasan A, DiDonato JA. Effects of resveratrol on gene expression in renal cell carcinoma. Cancer Biol Ther. 2004 Sep;3(9):882

9. Hauser P, Schwarz C, Mitterbauer C, Regele HM, Mühlbacher F, Mayer G, Perco P, Mayer B, Meyer TW, Oberbauer R. Genome-wide gene-expression patterns of donor kidney biopsies distinguish primary allograft function. Lab Invest. 2004 Mar;84(3):353-61

10. Arciero, C., Somiari, S. B., Shriver, C. D., Brzeski, H., Jordan, R., Hu, H., Ellsworth, D. L. and Somiari, R. I. (2003). Functional relationship and gene ontology classification of breast cancer biomarkers. Int. J. Biol. Markers 18, 241-272.

11. Fliser D, Novak J, Thongboonkerd V, Argilés A, Jankowski V, Girolami MA, Jankowski J, Mischak H. Advances in urinary proteome analysis and biomarker discovery. J Am Soc Nephrol. 2007 18(4):1057-71.

12. Casadio R, Martelli PL, Pierleoni A. The prediction of protein subcellular localization from sequence: a shortcut to functional genome annotationBrief Funct Genomic Proteomic. 2008 Feb 18 [Epub ahead of print

13. Chen XW, Liu M, Ward R.Protein Function Assignment through Mining Cross-Species Protein-Protein Interactions. PLoS ONE. 2008 Feb 6;3(2):e1562.

14. Camon EB, Barrell DG, Dimmer EC, Lee V, Magrane M, Maslen J, Binns D, Apweiler R An evaluation of GO annotation retrieval for BioCreAtIvE and GOA BMC Bioinformatics. 2005 ; 6 Suppl 1: S17.

15. Biswas M, O'Rourke JF, Camon E, Fraser G, Kanapin A, Karavidopoulou Y, Kersey P, Kriventseva E, Mittard V, Mulder N, Phan I, Servant F, Apweiler R. Applications of InterPro in protein annotation and genome analysis.Brief Bioinform. 2002 Sep;3(3):285-95.

16. Mulder N, Apweiler R.InterPro and InterProScan: tools for protein sequence classification and comparison. Methods Mol Biol. 2007;396:59-70.

17. Lovering RC , Dimmer E, Khodiyar VK, Barrell DG, Scambler, P, Hubank, M, Apweiler R, Talmud, PJ. 2008 Cardiovascular GO Annotation Initiative Year 1 Report: Why Cardiovascular GO? Proteomics (in press)

Ontologies: An Introduction

Claudia d'Amato, Nicola Fanizzi, and Floriana Esposito

Dipartimento di Informatica, Universit`a degli Studi di Bari

{claudia.damato|fanizzi|esposito}@di.uniba.it

Abstract. The purpose of this paper is to give an introduction to the notion of ontology. Several meanings of the term "ontology" will be analysed and the current one (conceptualization of a certain domain) will be examined. Many languages can be adopted for representing an ontology. Here the Description Logics will be focused since they are the theoretical foundation of the OWL language, that is the de facto standard in the Semantic Web context. Reasoning on ontologies will be also considered, since it makes explicit knowledge that is implicitly asserted in the ontology. Hence the multiple uses of ontologies and reasoning on ontologies will be analysed.

Introduction

The term *Ontology* has its origin in philosophy. It refers to that branch of philosophy which deals with the nature and the organization of reality. In this sense Ontology tries to answer to the question: What is being? or: What are the features common to all beings? Following this definition, the term Ontology is usually contrasted with the term Epistemology which represents the philosophical branch that deals with the nature and the sources of knowledge [1].

In the last two decades, the term ontology has been largely adopted and used in several fields of computer science and information science, leading to a proliferation of definitions [2]:

1. Ontology as a specific "*syntactic*" object, namely as a concrete artifact at the syntactic level, to be used for a given purpose
   (a) Ontology as representation of conceptual a system via a logical theory
   (b) Ontology as the vocabulary used by a logical theory
   (c) Ontology as a meta-level specification of a logical theory
2. Ontology as a conceptual "*semantic*" entity (either formal or informal), namely as a conceptual framework at the semantic level
3. Ontology as specification of a *conceptualization*, namely as an intensional semantic structure which encodes the implicit rules constraining the structure of a piece of reality.

As result of these many definitions, several kinds of ontologies have been developed. They can be summarized in two main types. Indeed, definition 1 refers to a formalization of a specific knowledge referring to a particular domain of interest e.g. the biological domain, the social domain, medical, etc. These ontologies are called *Material Ontologies* [3] or Domain Ontologies. They define the particular meaning of the terms as they are intended in the considered domain. For instance, an ontology referring to the *poker* domain would model the concept of *playing card*, while an ontology referring to the *computer hardware domain* would model the concepts *punch card* and *video card*. Examples of Domain Ontologies are: the Chemicals Ontology [4] and the Gene Ontology [5].

Definitions 2 and 3 refer to a conceptualization reflecting the study of the organisation and the nature of the world, independently of the form of our knowledge about it, namely they refer to the systematic, formal, axiomatic development of the logic of all forms and modes of being. These ontologies are called Formal Ontologies [3, 6] or General Ontologies or Upper Ontologies or Foundation Ontologies. What a formal ontology is concerned in, is not so much the bare existence of certain individuals, rather the rigorous description of their forms. In practice, a formal ontology can be intended as the theory of a priori distinctions: 1) among the entities of the world (such as physical objects, events, regions, quantities of matter...); 2) among the metalevel categories used to model the world (such as concepts, properties, qualities, states, roles, parts...). It describes the basic concepts and relationships invoked when information about any domain is expressed in natural language and refers to a model common to a set of objects that is generally applicable across a wide range of domain ontologies. Differently from the domain ontologies, where concepts and terms are generally defined w.r.t. the particular domain, upper level ontologies usually contain a core glossary whose terms can be used to describe a set of domains. Examples of upper level ontologies are: DUBLIN CORE [7], the General Formal Ontology (GFO) [8], OpenCys/ResearchCys [9], SUMO [10] and DOLCE [11].

Ontologies can also vary in their structure, spanning from simple lexicons, controlled vocabularies, categorically organized thesauri and taxonomies (e.g. WordNet[1]) to full-blown ontologies where properties can define new concepts and where concepts name relationships (e.g. the Wine Ontology[2]).

Currently, most of researchers (especially in the field of Computer Science) consider the Gruber's ontology definition: "*ontologies are formal specification of a shared conceptualization*" [12] and the one extended by A. Gomez Perez et al. [13] "*An ontology is a formal conceptualization of a domain that is shared and reused across domains, tasks and group of people*" as the most appropriate definitions. From them, the current role of ontologies in computer science and information science is straightforward [14]: making the semantics explicit so that: 1) ontologies can represent the common agreement of a certain community and they can be considered as a knowledge reference for that community; 2) ontologies can make possible the sharing of the consistent understanding of what an information means; 3) ontologies can make possible the increasing of the interoperability among different information systems.

How to write an ontology

An ontology is a knowledge base formalizing the common understanding of a certain domain[3]. Basics elements of an ontology are [15]:

Individuals: are the "ground level" components of an ontology. They can be of two different kinds: 1) *concrete objects of a domain* (such as people, animals, automobiles, molecules etc.); 2) *abstract individuals* (such as number and words).

Concepts: are collections of objects. They may contain individuals, other classes, or a combination of both. Some examples of concepts (sometimes also called classes) are: the concept Molecule representing the set of all molecules, the concept `Vehicle` representing the set of all vehicles, the concept Car representing the set of all cars, for instance Fiat Punto, Lamborghini, Ferrari etc.

Attributes: whose role is to describe the objects in the ontology. An attribute has at least a *name* and a *value* and it is used to specify information that refer to the particular object to which the attribute is attached to. For instance, given the individual Ford Explorer the attribute name Number-of-doors can be specified with value 4, in the same way the attribute Transmission can be specified with value 6 − *speed*.

Relationships: whose role is to make explicit the links between objects. A relation can be model as: 1) an attribute whose value is another individual in the ontology. For instance, given the individuals Ford Explorer and Ford Bronco, the attribute Successor: Ford Explorer of Ford Bronco means that Explorer is the model that replaced Bronco. 2) A mathematical relation. For instance Successor(Ford Bronco,Ford Explorer).

Much of the power of ontologies comes from the ability to describe relationships, since the whole relation set describes the semantics of the considered domain.

In order to represent an ontology, several ways have been used [16]: 1) the informal way characterized by the use of the natural language; 2) the semi-formal way, in which only limited structured forms of the natural language are allowed; 3) the formal way characterized by the use of a formal language having a formal semantics.

Since ontologies represent the shared and common understanding of a certain domain, they need to be represented by means of a language

[1] `http://wordnet.princeton.edu/`

[2] `http://www.w3.org/TR/owl-guide/wine.rdf`

[3] Note that this intuitive definition of an ontology is valid both for upper level and domain ontologies. Indeed in both cases a set of aspects pertaining a particular domain of interest is focused. The main difference is given by the way in which they are formalized.

that is non-unbiguos and universally understandable.

In the last few years several representation languages have been proposed: 1) *CycL*[4] (developed in the Cyc project[5]) that is a language based on first-order predicate calculus with some higher-order extensions; 2) *RIF*[6] (Rule Interchange Format) combining ontologies and rules (expressed in F-Logic) in order to make possible their interchange on the Semantic Web; 3) *KIF*[7] (developed in a DARPA project) that is a language inspired to first-order logic with a syntax based on S-expressions and designed for the use in the interchange of knowledge among disparate computer systems; 4) *OWL*[8] that is the Web Ontology Language intended to be used over the World Wide Web developed as a follow-on from RDF[9] and RDFS[10], and earlier ontology language projects such as OIL, DAML, DAML+OIL. It is supported by the well-founded semantics of *Description Logics* (henceforth DLs) [17], together with a series of available automated *reasoning services* allowing to derive logical consequences from an ontology. Here, the basics of DLs will be analyzed and how DL concepts and roles can be expressed in OWL language will be shown.

In DLs, descriptions are inductively defined starting with a set N_C of *primitive concept* names and a set N_R of *primitive roles*. The semantics of the descriptions is defined by an *interpretation* $I = (\Delta^I, \cdot^I)$, where Δ^I is a non-empty set representing the *domain* of the interpretation, and $\cdot^I$ is the *interpretation function* that maps each $A \in N_C$ to a set $A^I \subseteq \Delta^I$ and each $R \in N_R$ to $R^I \subseteq \Delta^I \times \Delta^I$. The *top* concept ⊤ is interpreted as the whole domain Δ^I, while the *bottom* concept ⊥ corresponds to ∅;. Complex descriptions can be built using primitive concepts and roles and the constructors showed in Tab. 1, whose semantics is also specified. Depending from the subset of constructors that is considered, several DL with different expressive power are obtained (see [17]).

Name	Syntax	Semantics
atomic negation	$\neg A,\ A \in N_C$	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
full negation	$\neg C$	$C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
concept conj.	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
concept disj.	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
full exist. restr.	$\exists R.C$	$\{a \in \Delta^{\mathcal{I}} \mid \exists b\, (a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}$
universal restr.	$\forall R.C$	$\{a \in \Delta^{\mathcal{I}} \mid \forall b\, (a,b) \in R^{\mathcal{I}} \rightarrow b \in C^{\mathcal{I}}\}$
at most restr.	$\leq nR$	$\{a \in \Delta^{\mathcal{I}} \mid \mid \{b \in \Delta^{\mathcal{I}} \mid (a,b) \in R^{\mathcal{I}}\} \mid \leq n$
at least restr.	$\geq nR$	$\{a \in \Delta^{\mathcal{I}} \mid \mid \{b \in \Delta^{\mathcal{I}} \mid (a,b) \in R^{\mathcal{I}}\} \mid \geq n$
qualif. at most r.	$\leq nR.C$	$\{a \in \Delta^{\mathcal{I}} \mid \mid \{b \in \Delta^{\mathcal{I}} \mid (a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\} \mid \leq n$
qualif. at least r.	$\geq nR.C$	$\{a \in \Delta^{\mathcal{I}} \mid \mid \{b \in \Delta^{\mathcal{I}} \mid (a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\} \mid \geq n$
one-of	$\{a_1, a_2, ...a_n\}$	$\{a \in \Delta^{\mathcal{I}} \mid a = a_i, 1 \leq i \leq n\}$
has value	$\exists R.\{a\}$	$\{b \in \Delta^{\mathcal{I}} \mid (b, a^{\mathcal{I}}) \in R^{\mathcal{I}}\}$
inverse of	R^-	$\{(a,b) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid (b,a) \in R^{\mathcal{I}}\}$

Table 1. DL Constructors semantics.

An ontology, namely a *knowledge base* $K = \langle T, A \rangle$ contains two components: a *T-box T* and an *A-box A*. T is a set of concept definitions $C \equiv D$, meaning $C^I = D^I$, where C is the concept name and D its description, given in terms of the language constructors[11]. A contains extensional assertions on concepts and roles, e.g. $C(a)$ and $R(a, b)$, meaning, respectively, that $a^I \in C^I$ and $(a^I, b^I) \in R^I$; $C(a)$ and $R(a, b)$ are said respectively instance of the concept C and instance of the role R, more generally it is said (without loss of generality) that the individual a is instance of the concept C and the same for the role.

In the following, an example of knowledge base describing the family domain is reported.

Let NC = {Female, Male, Human} be the set of primitive concepts and let NR = {HasChild, HasParent, HasGrandParent, HasUncle} be the set of primitive roles. By the use of some constructors showed in Tab. 1, namely concept conjunction and disjunction, full existential restriction and at least number restriction, the following complex concept descriptions can be defined.

T = { Woman ≡ Human ⊓ Female;
Man ≡ Human ⊓ Male;
Parent ≡ Human ⊓ ∃HasChild.Human;
Mother ≡ Woman ⊓ Parent;
Father ≡ Man ⊓ Parent;
Child ≡ Human ⊓ ∃HasParent.Parent;
Grandparent ≡ Parent ⊓ ∃HasChild.(∃ HasChild.Human);
Sibling ≡ Child ⊓ ∃HasParent.(∃ HasChild ≥ 2);
Niece ≡ Human ⊓ ∃HasGrandParent.Parent ⊔ ∃HasUncle.Uncle;
Cousin ≡ Niece ⊓ ∃HasUncle.(∃ HasChild.Human) }

The first concept description introduces a new concept name Woman as someone that is a

[4] `http://www.cyc.com/cycdoc/ref/cycl-syntax.html`

[5] `http://www.cyc.com/`

[6] `http://www.w3.org/2005/rules/wiki/RIF Working Group`

[7] `http://logic.stanford.edu/kif/kif.html`

[8] `http://www.w3.org/2004/OWL/`

[9] `http://www.w3.org/RDF/`

[10] `http://www.w3.org/TR/rdf-schema/`

[11] Inclusion axioms of kind $C \sqsubseteq D$ are also allowed in a TBox as partial definitions, anyway they will be not considered here. See [17] for more details about them.

human female. The seventh concept definition introduces a new concept Grandparent that is defined as a parent that has a human child and this human child has a human child as well. The eighth description introduce the concept Sibling as someone that has a parent having at least two children.

The TBox represents the conceptualization of the considered domain. In order to have the effective knowledge of the particular domain, the ABox has to be specified. In the following, the Abox A corresponding to the TBox specified above and focusing on a particular set of families is reported.

$\mathcal{A}$ = { Woman(Claudia), Woman(Tiziana), Father(Leonardo), Father(Antonio),
Father(AntonioB), Mother(Maria), Mother(Giovanna), Child(Valentina),
Sibling(Martina), Sibling(Vito), HasParent(Claudia,Giovanna),
HasParent(Leonardo,AntonioB), HasParent(Martina,Maria),
HasParent(Giovanna,Antonio), HasParent(Vito,AntonioB),
HasParent(Tiziana,Giovanna), HasParent(Tiziana,Leonardo),
HasParent(Valentina,Maria), HasParent(Maria,Antonio),
HasSibling(Giovanna,Maria), HasSibling(Vito,Leonardo),
HasSibling(Tiziana,Claudia), HasSibling(Valentina,Martina),
HasChild(Leonardo,Tiziana), HasChild(Antonio,Giovanna),
HasChild(Antonio,Maria), HasChild(Giovanna,Tiziana), HasChild(Giovanna,Claudia),
HasChild(AntonioB,Vito), HasChild(AntonioB,Leonardo), HasChild(Maria,Valentina),
}

From this Abox we know that Tiziana is a woman and she has a sibling which is Claudia and a parent that is Leonardo.

In order to work with Ontology Languages, there are some useful tools like Ontology Editor, Ontology DBMS (to store and query an ontology) and Ontology Warehouse (to integrate and explore a set of related ontologies). Particularly, by the use of an ontology editor such as Prot´eg´e[12], concepts are described by the use of DLs and automatically saved in OWL format[13]. This allows to make the created ontologies available and interoperable on the Web. In the following, part of the TBox and ABox specified in OWL language is reported.

```
<owl:Class rdf:ID="Human"/>
<owl:Class rdf:ID="Father">
 <owl:equivalentClass>
  <owl:Class>
   <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:ID="Man"/>
    <owl:Class rdf:ID="Parent"/>
   </owl:intersectionOf>
  </owl:Class>
 </owl:equivalentClass>
</owl:Class>
<owl:Class rdf:ID="Female">
 <owl:disjointWith>
  <owl:Class rdf:ID="Male"/>
 </owl:disjointWith>
</owl:Class>
<owl:Class rdf:ID="Child">
 <owl:equivalentClass>
  <owl:Restriction>
   <owl:someValuesFrom>
    <owl:Class rdf:about="#Parent"/>
   </owl:someValuesFrom>
   <owl:onProperty>
    <owl:ObjectProperty rdf:ID="HasParent"/>
   </owl:onProperty>
  </owl:Restriction>
 </owl:equivalentClass>
 <rdfs:subClassOf rdf:resource="#Human"/>
</owl:Class>
```

Reasoning on Ontologies

As observed in the previous section, an ontology can be seen as a set of axioms (given by the concept definitions) and assertions (given by the concept and role instances). As such, it contains implicit knowledge. In order to make it explicit, logic inferences, based on deductive reasoning, can be performed. For accomplishing this goal, several reasoner, based on the DL formal semantics have been developed, such as *FaCT*[14], *FaCT++*[15], *RACER*[16] and *PELLET*[17]. They implement the well known standard inference services available in DLs and that are analyzed in the following.

In the construction of a TBox $\mathcal{T}$, new concepts are defined, likely in terms of concepts already defined. During this process, it is important to find out whether a newly defined concept makes sense or whether it is contradictory. Logically, a concept makes sense if there is an interpretation that satisfies the axioms of $\mathcal{T}$ (that is, if there is a model of $\mathcal{T}$). A concept with this property is said to be *satisfiable* w.r.t. $\mathcal{T}$ and *unsatisfiable* otherwise. Checking satisfiability of concepts is formally defined as:

[12] http://protege.stanford.edu/
[13] This is possible since DLs represent the theoretical foundation of OWL.

[14] http://www.cs.man.ac.uk/ horrocks/FaCT/
[15] http://owl.man.ac.uk/factplusplus
[16] http://www.sts.tu-harburg.de/ r.f.moeller/racer/
[17] http://pellet.owldl.com/

Definition 3.1 (Satisfiability). A *concept C is satisfiable w.r.t. T if there exists a model I of T such that C^I is nonempty. In this case it is said also that I is a model of C.*

Example 3.1 (Concept satisfiability). Let T = { Parent, Man, Woman ≡ ¬ Man, Mother ≡ Woman ⊓ Parent} be the TBox of reference and let Man ⊒ Mother a new axiom added to the TBox. It is straightforward to see that the new axiom is *unsatisfiable* w.r.t TBox because the disjointness constraint between Man and Woman is violated.

Other important inferences for concepts are: *subsumption*, checking *concepts equivalence* and checking *concepts disjointness*. They can be formally defined as follow.

Definition 3.2 (Subsumption). *A concept C is subsumed by a concept D w.r.t. T if $C^I \subset D^I$ for every model I of T. In this case it is written $C \sqsubseteq_T D$* or *$T \models C \sqsubseteq D$.*

The subsumption test is the most common inference in DL. It tests if a concept D is more general than another concept C. One of its most useful use cases is building up concepts subsumption hierarchies (taxonomies) in order to organize the concept graph of a TBox, for instance in a knowledge base management system.

Looking at the TBox reported is Sect. 2, it is straightforward to see that the concept Parent subsumes the concepts Mother and Father.

Definition 3.3 (Equivalence). *Two concepts C and D are equivalent w.r.t. T if $C^I = D^I$ for every model I of T. In this case it is written $C \equiv_T D$ or $T \models C \equiv D$.*

Definition 3.4 (disjointness). *Two concepts C and D are disjoint w.r.t. T if $C^I \cap D^I = \emptyset$ for every model I of T.*

Besides of TBox inference services, ABox inference services are also available. They are useful in order to make explicit assertional knowledge implicitly contained in the ABox. Standard reasoning tasks for ABoxes are: 1) ABox consistency check; 2) Instance checking; 3) Retrieval.

ABox *consistency check* solves the problem of checking if a new assertion (concept or role assertion) in an ABox A makes A inconsistent or not w.r.t. the TBox T. This service is important because the representation of the knowledge in the ABox (after a TBox T has been built and TBox taxonomy and consistency have been checked) has to be consistent with T, otherwise arbitrarily conclusions can be drawn from it. This means that, considering a simple TBox T = {Woman ≡ Person ⊓ Female, Man Person ≡ ¬Female}, if the ABox contains the assertions Woman(MARY) and Man(MARY), the system should be able to find out that, together with T, these statements are inconsistent due to disjointness axiom in T for which an individual cannot be instance of both Man and Woman. Formally, it is said that:

Definition 3.5 (ABox Consistency (w.r.t. a TBox)). *An ABox A is consistent with respect to a TBox T if there exists an interpretation that is a model of both A and T.*

Note that, if instead of the T = {Woman ≡ Person ⊓ Female, Man ≡ Person ⊓ ¬Female} the TBox T' = {Woman,Man} is considered, the assertions Woman(MARY) and Man(MARY) are consistent w.r.t. T' since there are no restrictions imposed on the interpretation of Woman and Man in T'.

The *instance check* is the prototypical ABox inference consisting in checking whether an assertion is entailed by an ABox. It is used in order to allow queries concerning concepts, roles and individuals over an ABox. Formally:

Definition 3.6 (Assertion entailment). *An assertion α is entailed by an ABox A w.r.t. a TBox T and it is written $A \models T\ \alpha$, if every interpretation that satisfies A w.r.t. T, (i.e. every model of A w.r.t. T), also satisfies w.r.t. T.*

Looking at the TBox in Sect. 2, we know, from the knowledge base that Claudia is instance of the concept Woman. It is possible to entail from the knowledge base (and specifically from the role assertions HasSibling(Tiziana,Claudia) and Has-Parent(Tiziana,Leonardo)) that Claudia is also instance of the concept Sibling.

Since a knowledge base is also a way for storing information about individuals, it is likely to require to know, for instance, all individuals that are instances of a given concept description C. This is equivalent to use the description language to formulate queries on the knowledge base. The possibility of making queries to a knowledge base for getting the set of individuals that are instances of a certain concept is called retrieval problem. The *retrieval problem* can be formally defined as:

Definition 3.7 (Retrieval Problem). *Given an ABox A and a concept C, finds all individuals a such that $A \models_T C(a)$*

A straightforward algorithm for performing the retrieval problem can be realized by testing for each individual occurring in the ABox whether it is an instance of the query concept C.

An important aspects that need to be considered for modeling an ontology and also for reasoning is the *Open World Assumption* (OWA) that is made by DLs. It is opposite to the *Closed World Assumption* (CWA) generally made in Data Base setting. The difference is crucial and strongly affects reasoning and representation. While in the CWA the absence of information is interpreted as negative information, in the OWA the absence of information is interpreted as unknown information. Let us consider the following example. Let $T = \{$ Female,Woman$\}$ be a TBox of reference and $A = \{$ Female(Ann),Woman(Sara)$\}$ the corresponding ABox. If the CWA is assumed, by querying the knowledge base with $q =$ Female(Sara), namely by asking if Sara is instance of the concept Female, the reply of the reasoner will be not, since neither there is any assertion of kind Female(Sara) in the ABox nor this assertion can be derived, since both Female and Woman are primitive concepts and they are subconcepts only of the $\top$ concept. On the contrary, if the OWA is assumed, by querying the knowledge base with $q =$ Female(Sara), the reasoner will not give any reply, since neither the assertion is in the ABox not it can be derived. When an ontology is modelled, the OWA has to be taken into account.

Using DL Inference Operators

As seen in the previous sections, knowledge representation and reasoning are strictly related. Depending on the information available and its correctness implicit knowledge can be derived from the knowledge base. Anyway, the ability in performing reasoning also depends on the expressive power of the adopted representation language. Indeed, with the increasing of the expressive power of the representation language the computational complexity of the reasoning procedures increases as well. This provokes not only a high computational time but also that reasoning procedures can become semidecidable. Namely, it can happen that if the query forwarded to a reasoner is not a conclusion that can be proved from the logical premises of the available axiom set, the reasoning procedure cannot termine. In order to cope with this aspect, OWL language provides three increasingly expressive sub-languages (see W3C Recommendation `http://www.w3.org/2004/OWL/` for more details):

OWL Lite : *decidible* with desirable computational properties. It supports the classification hierarchy inference and simple constraint features, i.e. cardinality constraints on properties where only cardinality values of 0 and 1 are permitted.

OWL DL : *decidible* but subject to higher worst-case complexity. It allows restrictions such as type separation, namely a class cannot also be an individual or a property, as well as a property cannot also be an individual or a class.

OWL Full : not *decidible*. It allows that a class can be treated simultaneously as a collection of individuals and as an individual in its own right.

Reasoning services play a key role in the ontology life-cycle. They are largely used to support the knowledge engineer during the ontology design task. In this phase the the *concept satisfiability check* is particularly important. It is used to ensure that a new added concept does not make the knowledge base inconsistent or does not imply unexpected relationships.

Another important phase in the ontology life-cycle is the ontology mapping and alignment [18] which mainly consist in obtaining a common knowledge base, given two or more ontologies autonomously developed. In this phase a reasoner is generally exploited to compute an integrated concept hierarchy and for checking its consistency.

During the ontology deployment, Abox reasoning services are used. Specifically, the Abox consistency check is used in order to determine if the set of facts asserted in the ABox are consistent w.r.t. TBox. *Instance checking* and *retrieval* are used for querying the knowledge base.

In order to support such necessary operations for the ontology life-cycle, several ontology tools have been developed. Some examples are: Prot´eg´e[18], a free, open source ontology editor and knowledge-base framework; Chimaera[19], a system for creating and maintaining distributed ontologies on theWeb; Ontolingua[20], a distributed collaborative environment to browse, create, edit, modify, and use ontologies; OntoEdit[21], an engineering environment for the development and maintenance of ontologies using graphical means;WebOnto[22], a Java applet coupled with a customised web server allowing to browse and edit knowledge models over the Web; KAON[23], an open-source infrastructure for ontology creation and management, and providing a framework for building ontology-based applications.

Besides of the standard inference services presented in Sect. 3, also non-standard inference services have been developed (see [19] for more details) in order to support ontology building and maintenance and for getting information from them. Among the most useful non-standard inference services there are the computation of: the *least common subsumer*, the *most specific concept*, concept *matching/unification*, *concept rewriting*. Each of them has been introduced to solve a particular problem. Indeed, standard inference services check for new concepts that are manually defined, but they do not directly support the process of actually defining new concepts. The computation of the *least common subsumer* and *most specific concept* try to overcome this problem. Moreover, if a knowledge base is maintained by different knowledge engineers, it is necessary to have a tool for detecting multiple definitions of the same intuitive concept, since different knowledge engineers might use different names for the "same" primitive concept. In order to accomplish this problem, the standard equivalence test may not be adequate. The non-standard inference service performing *unification of concept descriptions* tackles this problem by allowing to replace concept names by appropriate concept descriptions before testing for equivalence. Concept *matching* is a special case of unification. It is used for pruning irrelevant parts of large concept descriptions before displaying them to the user. Furthermore, the non-standard inference service performing the *rewriting of concept descriptions* allows to transform a given concept description C into a "better" description D satisfying certain optimality criteria (e.g., small size) and that is related (e.g., by equivalence or subsumption) to the original description C.

Besides of the purely deductive based approach for performing reasoning, inference services grounded on the inductive approach and Machine Learning techniques have also been developed. In [20, 21], a classifier for performing inductive concept retrieval and query answering is presented. It allows to induce new concept assertions that are not logically derivable. Such assertions can be suggested to the knowledge engineer in order to make semi-automatic the time consuming ontology population task. In [22], a clustering method for performing concept drift and novelty detection is presented. Other effort have been dedicated for making semi-automatic the ontology learning task starting from different source of information such as texts [23] or examples [24, 25].

Conclusions

An overview of the meaning and usage of ontologies has been presented and the way in which they can be described has been showed.

Ontologies represent a formal domain conceptualization that is shared and reused across domains, tasks and group of people. Different kinds of ontologies, namely upper level or domain ontologies have been discussed. Independently from the kind of ontologies, their main functions

[18] http://protege.stanford.edu/
[19] http://ksl.stanford.edu/software/chimaera/
[20] http://www.ksl.stanford.edu/software/ontolingua/
[21] http://www.ontoknowledge.org/tools/ontoedit.shtml
[22] http://kmi.open.ac.uk/projects/webonto/
[23] http://kaon.semanticweb.org/

are: constituting a community reference; sharing consistent understanding of what information means; making possible interoperability between systems; making the Web machine-readable and processable besides of human-readable (Semantic Web).

Reasoning on ontologies has been also discussed. It represents a crucial aspect in managing ontologies since it allows to make explicit the knowledge that is implicitly asserted in an ontology. Several reasoning operators have been analyzed and the current lines of research in performing reasoning for making the ontology life-cycle semiautomatic have also been considered.

References

[1] Dolce : a descriptive ontology for linguistic and cognitive engineering. http://www.loacnr.it/DOLCE.html.

[2] The dublin core metadata initiative. http://dublincore.org/.

[3] The gene ontology. http://www.geneontology.org/.

[4] The general formal ontology. http://www.onto-med.de/en/theories/gfo/index.html.

[5] Opencyc. http://www.opencyc.org/.

[6] The suggested upper merged ontology (sumo). http://www.ontologyportal.org/.

[7] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors. The Description Logic Handbook. Cambridge University Press, 2003.

[8] M. Bunge, editor. Treatise on basic philosophy. Ontology I: the furniture of the world,. Dordrecht, Reidel, 1977.

[9] N. Choi, I. Y. Song, and H. Han. A survey on ontology mapping. SIGMOD Rec., 35(3):34–41, 2006.

[10] N. B. Cocchiarella. Formal ontology. In H. Burkhardt and B. Smith, editors, Handbook of Metaphysics and Ontology., pages 640–647. Philosophia Verlag, Munich, 1991.

[11] C. d'Amato, N. Fanizzi, and F. Esposito. Query answering and ontology population: an inductive approach. In S. Bechhofer et al., editor, Proc. of the European Semantic Web Conference, ESWC2008, LNCS. Springer, 2008.

[12] F. Esposito, N. Fanizzi, L. Iannone, I. Palmisano, and G. Semeraro. Knowledge-intensive induction of terminologies from metadata. In F. van Harmelen, S. McIlraith, and D. Plexousakis, editors, ISWC2004, Proceedings of the 3rd International Semantic Web Conference, volume 3298 of LNCS, pages 441–455. Springer, 2004.

[13] N. Fanizzi, C. d'Amato, and F. Esposito. Conceptual clustering and its application to concept drift and novelty detection. In S. Bechhofer et al., editor, Proceedings of the European Semantic Web Conference, ESWC2008, LNCS. Springer, 2008.

[14] H. J. Feldman, M. Dumontier, S. Ling, and C.W.V. Hogue. Co: A chemical ontology for identification of functional groups and semantic comparison of small molecules. In FEBS Letters., volume 579, pages 4685–4691. 2005.

[15] A. G´omez-P´erez and V.R. Benjamins. Overview of knowledge sharing and reuse components: Ontologies and problem-solving methods. In Proc. of Ontology and Problem-Solving Methods: Lesson learned and Future Trends, Workshop at IJCAI, volume 18 of CEUR Workshop Proceeding, pages 11–15. CEUR, 1999.

[16] A. G´omez-P´erez and M. D. Rojas-Amaya. Ontological reengineering for reuse. In Knowledge Acquisition, Modeling and Management, volume 1621 of LNCS, pages 139–156. Springer, 1999.

[17] T.R. Gruber, editor. A translation approach to portable ontology specifications. 1993.

[18] N. Guarino and P. Giaretta. Ontologies and knowledge bases: Towards a terminological clarification. In N. Mars, editor, Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing., pages 25–32. IOS Press, 1995.

[19] J.-U. Kietz and K. Morik. A polynomial approach to the constructive induction of structural knowledge. Machine Learning, 14(2):193–218, 1994.

[20] A. Maedche and S. Staab. Ontology learning. In S.Staab and R. Studer, editors, Handbook on Ontologies, International Handbooks on Information Systems, pages 173–190. Springer, 2004.

[21] C. d'Amato. N. Fanizzi. Inductive concept retrieval and query answering with semantic knowledge bases through kernel methods. In Proceedings of the the 11th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems, KES 2007, LNcs, Vietri sul Mare, Italy, 2007. Springer.

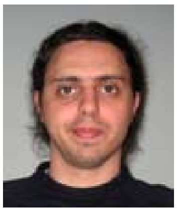
[22] R. K¨usters. Non-Standard Inferences in Description Logics: From Foundations and Definitions to Algorithms and Analysis, volume 2100 of LNCS/LNAI. Springer, 2001.

[23] S. Shapiro, editor. Encyclopedia of Artificial Intelligence. John Wyley, 1987.

[24] J. F. Sowa, editor. Knowledge Representation: Logical, Philosophical, and Computational Foundations. Brooks Cole Publishing Co., 2000.

[25] Steffen Staab and Rudi Studer, editors. Handbook on Ontologies. International Handbooks on Information Systems. Springer, 2004.

Enhancing Gene Ontology Annotation through Collaborative Tagging



Domenico Gendarmi[1]**, Andreas Gisel**[2]**, Filippo Lanubile**[1]

[1] University of Bari, Dipartimento di Informatica, Bari, Italy. [2] Consiglio Nazionale delle Ricerche, Istituto di Tecnologie Biomediche, Bari, Italy.

Introduction

Ontologies play a relevant role in providing a common understanding of a domain to share knowledge among human beings and software agents [6]. The domain model implicit in an ontology can be taken as a unified structure for giving information a common representation and semantic [1]. However the ontology development process is typically led by single or small groups of experts, with users mostly playing a passive role. Such an elitist approach in building ontologies leads to several limitations that hinder the primary purpose of large-scale knowledge sharing.

The achievement of a widespread participation in the ontology development process is often hampered by entry barriers, like the lack of easy-to-use and intuitive tools for ontology contribution. Another relevant problem is the temporal extent of reliable knowledge which tends to be short. More information users learn, more the agreement and consensus among them evolve; thus new pieces of knowledge have to be committed and older pieces have to be constantly checked and validated. However, current ontologies require that all the changes have to be captured and introduced by the same knowledge engineers who created them. To be really effective, ontologies thus need to change as fast as the parts of the world they describe [7]. Such an efficiency, however, can only be achieved by directly interacting with the proper community. Community participation to ontology development has already been identified as a solution to a more complete and up-to-date structured knowledge construction [13].

In this paper we introduce an approach to knowledge evolution which aims to exploit the ability of collaborative tagging in fostering community participation to increase the speed of adjournment of an initial knowledge structure. Participants can organize some piece of knowledge according to a self-established vocabulary, building up personal taxonomies for searching and browsing through their own information spaces. By sharing portions of their knowledge, users can also create connections and negotiate meaning with people having similar interests. The main goals of the proposed approach are: (1) to allow users to organize personal information spaces, starting from a prearranged knowledge structure; and (2) to take advantage of users' contribution for better reflecting the community evolution of a shared knowledge structure.

Gene Ontology context

Gene Ontology (GO) [9] is the most widely accepted knowledge structure for the description of genes and their products in any organism. It can be regarded as a controlled and structured vocabulary divided in three independent knowledge spaces which allows the description of molecular functions, biological processes and cellular locations of any gene product of any organism. Within each of these ontologies, classes, describing biological aspects (terms), are organized in a tree like way, according to "is-part-of" and "is-a" relationships. The structure is well adapted to computational processing and is used for the functional annotations of a large amount of gene products, with detailed description from mainly model organisms and human gene products (Gene Ontology and Annotation, GOA) [10][11].

Typically, only a very small group is responsible for the maintenance of the ontology and this group is usually different from the actual users of the ontology which have no direct influence on the update and improvement of the ontology. In the GOA context, the situation is even more complex, since there are three groups involved: (1) the ontology curators, in charge of developing and maintaining the GO and its annotations; (2) the bioinformaticians, which use the GOA for the analysis of biological data; and (3) the scientists, which hold the knowledge and can discover new information in their laboratories.

The efforts of the Gene Ontology Consortium (GOC) fruited in a drastically enlargement and improvement of the knowledge space (GOA) of the genomics and proteomics. The GOA is widely used to analyse large output lists of high- throughput approaches such as expression microarray, to get an overview what possible processes or functions are involved in the studied biological samples. However, since mostly institutions covering the model organisms and the human genome take part in this annotation process, a vast amount of gene products from non-model organisms are not or only purely annotated mostly by bioinformatics approaches comparing less known gene products with well known. In addition, this partially separated annotation efforts within those institutions creates a certain synonymy; often a biological aspect can have several different classes in the GO, because the curator can choose a flavour of a description that is the most suitable for him. Biologists not part of the GOC have the possibility to propose new GO classes with its terminology and annotations, but this possibility is poorly used.

Increasing the number of annotations with experimental background and reducing the present synonymy would valuably increase the knowledge space of GOA and improve the needed quality for all applications using the GOA. Therefore, for a significant improvement of the GO it is very important that all information from any laboratory flow in one or another way into the improvement of the GO. To this end we are searching for new methodologies and technologies to merge, in a simple way, a wide 'semi-controlled' knowledge space of scientists in the laboratory with a highly organized knowledge structure such as the GO.

An example of knowledge space fully user-driven is the result of collaborative tagging system, often named as folksonomy [12]. If it would be possible to combine both system and use user generated content to frequently update and improve the GO this could be a drastic step forward in the quality of the GO knowledge. Within this project we propose a so called community-driven evolution based on collaborative tagging which we would like to apply on the GO to have the large available knowledge of genomics and proteomics. The overall aim is not to change the GO, but to create a second layer of dynamic knowledge on the top of the existing one, allowing users to profit of this "regulated" information and curators to enlarge the knowledge space much faster.

Collaborative Tagging Systems

One of the major obstacles hindering the widespread adoption of controlled vocabularies is the constant growth of available content which anticipates the ability of any single authority to create and index metadata. In such contexts collaborative tagging represents a potential solution to the vocabulary problem [3].

Collaborative tagging has emerged as a new social-driven annotation method, as it shifts the creation of metadata for describing web resources, from an individual professional activity to a collective endeavour, where every user is a potential contributor. Collaborative tagging systems allow people to organize a set of resources, annotating them with tags via a web-based interface. The activity of labelling is called tagging, as it consists of attaching one or more tags to the resource. This activity is accomplished individually, as each user of the system is free to choose the tags he wishes, with no restrictions. However, while using the system every one can see who else is participating to it by observing others' tagging activities. This tight feedback loop brings that asynchronous and asymmetrical collaboration which makes these systems socials [8]. The result of such a social activity is a collection of annotations, also called folksonomy.

Existing folksonomies can be discriminated according to the kind of resources they allow to annotate. The most popular example is delicious (`http://del.icio.us`), often defined as a social bookmarking system. However, despite of the different kind of items they allow to annotate. A collaborative tagging system can be generally modelled as a tripartite 3-uniform hypergraph as shown in Figure 1 [1].

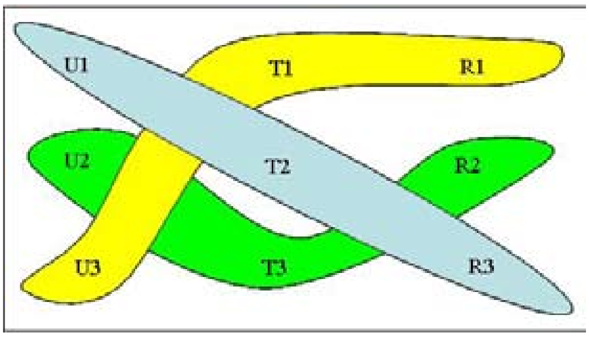


Figure 1. Hypergraph representing a folksonomy.

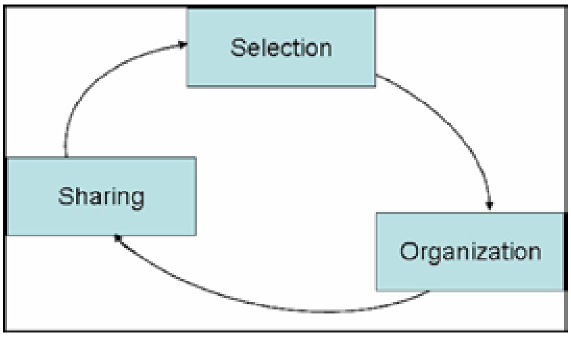


Figure 2. Three-step iteration process.

Collaborative tagging systems exhibit other interesting benefits such as their ability in adhering to the personal way of thinking. No forced restrictions on the allowed terms, as well as the lack of syntax to learn can shorten significantly the learning curve. Collaborative tagging systems also create a strong sense of community amongst their users, allowing them to realize how others have categorized the same resource or how the same tag has been used to label different resources. There is no need to establish a common agreement on the meaning of a tag because it gradually emerges with the use of the system. Marginal opinions can coexist with popular ones without disrupting the implicit emerging consensus on the meaning of the terms. The main drawbacks with tags concern semantic and cognitive issues, such as polysemy, synonymy and basic level variation [5].

Polysemy occurs when the same term is used for tags employed with different meanings. The polysemy problem affects query results by returning potentially related but often inappropriate resources. Polysemy is occasionally equalized to homonymy, however polysemous words have different meanings but related senses, while homonyms have multiple, unrelated meanings.

Synonymy takes place when different terms are used for tags having the same meaning. Synonymous tags are another source of ambiguity, severely hindering the discovery of all the relevant resources which are available in a tagging system. Polysemy and synonymy represent two critical aspects of a search, as they respectively affect precision and recall, which are typically used for evaluating information retrieval systems.

A further relevant problem, concerning the cognitive aspect of categorization, is the basic level variation of tags. Terms used to describe a resource can vary along a continuum of specificity ranging from very general to particularly specific. Different users can use terms at different levels of abstraction to describe the same resource, leading to a low recall in retrieving resources.

While it is well-known that search and retrieval are facilitated by structured subject headings, the tags which form a folksonomy are just flat terms. Besides the previous drawbacks, the lack of a structure is one of the main aspects which weaken severely the information retrieval in a collaborative tagging system.

Towards enhancing gene ontology annotation

In previous work we proposed an approach for applying collaborative tagging techniques to support the evolution of a knowledge structure adopted for the classification of a wide amount of digital resources [4].

According to our approach, a community of users collaborate for collectively evolving an initial knowledge following a three-step iteration process (Figure 2). A similar process could be applied to Gene Ontology context.

The first step, **Selection**, involves browsing and choosing genes or gene products to annotate. This step can be supported by existing tools that let users search for gene products and view the terms with which they are associate or alternatively browse through the Gene Ontology and finding genes related to GO terms (i.e. AmiGO, The GO Browser).

In the second step, **Organization**, scientists could create and organize their own private working space where to annotate the selected genes with GO terms (existing or new ones) they consider more appropriate.

It involves creating and structuring a personal information space according to individual interests. This step goes beyond current opportunities because it allows not only to store collections of genes of interest but also to group them using the desired GO terms.

The last step, **Sharing**, involves making public some selected gene products and corresponding terms. Sharing personal information about gene products among people or groups with similar research interests could evolve the knowledge about selected genes by many individuals in order to support a community knowledge evolution.

Final Remarks

Adopting a collaborative approach for ontology maintenance is a challenging research topic for the benefits it can bring to conventional approaches. Ontologies which are improved and used as a community will reflect the knowledge of users more effectively than ontologies maintained by knowledge engineers who struggle to capture all the variety taking place within a lively community.

The proposed approach can be regarded as a first step toward a collaborative system capable of allowing ontologies to evolve mainly through the contribution of its users. Personal information spaces could help scientists in laboratories to organize their own knowledge on gene products using their favourite terms, descriptions and annotations. Knowledge sharing among scientists with similar interests could create a feedback loop like in folksonomies. For example making public personal annotations let each scientist to discover gene products annotated with the same or similar term or conversely terms that have been used for the same gene product.

Finally, we argue that the GO could significantly benefit from this combination of 'semi-controlled' knowledge spaces of scientists in the laboratories and a central organized knowledge structure.

References

[1] Abbattista, F., Calefato, F., Gendarmi D., Lanubile, F.: Shaping personal information spaces from collaborative tagging systems. In B. Apolloni et al. (Eds.): KES 2007/ WIRN 2007, Part III, LNAI 4694, pp. 728–735, 2007.

[2] Davies, J., Fensel, D., and Harmelen, F. Towards the Semantic Web: Ontology-driven Knowledge Management. John Wiley & Sons, 2003.

[3] Furnas, G., Landauer, T., Gomez, L., and Dumais, S. The vocabulary problem in human-system communication, Communications of the ACM, 30, 11 (1987), 964-971.

[4] Gendarmi D., Abbattista F., Lanubile F.: Fostering knowledge evolution through community-based participation. In Proceedings of the Workshop on Social and Collaborative Construction of Structured Knowledge at WWW 2007, CEUR Workshop Proceedings, ISSN 1613-0073.

[5] Golder, S. and Huberman, B. Usage patterns of collaborative tagging systems, Journal of Information Science, 32, 2 (2006), 198-208.

[6] Gruber, T.: Toward Principles for the Design of Ontologies Used for Knowledge Sharing. International Journal Human-Computer Studies 43 (1993), 907-928.

[7] Haase, P., Völker, J., and Sure, Y. Management of dynamic knowledge, Journal of Knowledge Management, 9, 5 (2005), 97-107.

[8] Mathes, A. Folksonomies-Cooperative Classification and Communication Through Shared Metadata. Technical Report, LIS590CMC, Computer Mediated Communication, Graduate School of Library and Information Science, University of Illinois, Urbana-Champaign, 2004.

[9] The Gene Ontology Consortium: Gene Ontology: tool for the unification of biology. Nat Genet 2000, 25:25-29.

[10] The Gene Ontology Consortium: Creating the gene ontology resource: design and implementation. Genome Res 2001, 11:1425-1433.

[11] The Gene Ontology Consortium: The Gene Ontology (GO) project in 2006. Nucleic Acids Res 2006, 34: D322-D326.

[12] Vander Wal, T. Folksonomy Definition and Wikipedia.2005.

[13] Zhdanova, A. V., Krummenacher, R., Henke, J., and Fensel, D. 2005. Community-Driven Ontology Management: DERI Case Study. In Proceedings of the the 2005 IEEE/WIC/ACM international Conference on Web intelligence. 2005.

Functional profiling at CIPF: from Blast2GO to Babelomics, all strategies for every species

Ana Conesa

Bioinformatics Department,
Centro de Investigaciones
Prínicipe Felipe, Valencia, Spain

aconesa@cipf.es

Introduction

Over the last years, the application of functional genomics approaches has generalized in biological research. High-throughput technologies are no longer only accessible to the investigation of model systems, but practically any organism counts with a research project that aims at the characterization of its genome. Functional genomics means that the genes are investigated for their function, which implies that definitions for gene functions must also be available. The Gene Ontology (GO, `http://www.geneontology.org`; [1]) is without doubt the most extensive vocabulary for describing molecular functionalities and its wide use by the life sciences research community brings closer the utopia of a universal schema for the functional characterization of all known genes and gene products. But, obviously, functional annotation by the GO – or any other scheme - is only a means towards the understanding of the biology. Once a functional classification is available, analytical methodologies are required to derive knowledge from data. Therefore, for an effective functional genomics research two requirements need to be met:

- enough and high quality functional annotations must be available for the organism of study. This is readily available for model organisms in public repositories, but for less studied species, this information usually needs to be created;
- powerful statistical methods are required to analyse the experimental data.

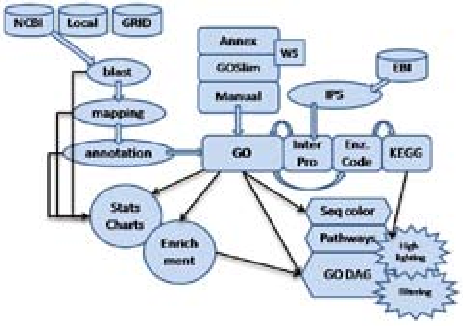


Figure 1. Schematic representation of Blast2GO application. GO annotations are generated through a 3 step process: blast, mapping, annotation. InterPro terms are obtained from InterProScan at EBI, converted and merged to GOs. GO annotation can be modulated from Annex, GOSlim web services and manual editing. EC and KEGG annotations are generated from GO. Visual tools include sequence colour code, KEGG pathways and GO graphs with node highlighting and filtering options. Additional annotation data-mining tools include statistical charts and gene set enrichment analysis functions.

Blast2GO (B2G, `http://www.blast2go.org`; [2,3]) is a versatile bioinformatics tool in the functional annotation field. With an average use of 2000 launches per month and being adopted as annotation engine in over 30 publications last year, it is one of the most world-wide used software for *de novo* assignment of functional terms. Babelomics (`http://www.babelomics.org`; [4,5]) is a complete suite for the function-based statistical analysis of genomics data. It contains different modules for analysing gene blocks, ranks, tissues and biomedical literature from a functional prospective. After a solid history of developments, it counts with more than 400 supporting citations.

These two suites are now working together at the Centro de Investigaciones Príncipe Felipe CIPF (`http://bioinfo.cipf.es`) to provide a unique site of bioinformatics resources for the functional genomics study of virtually any organism. In this report we describe the main features of both Blast2GO and Babelomics and provide practical insights in the use of functional information to analyse genomics data.

The Blast2GO application

Figure 1 shows the basic components of the Blast2GO suite. Functional assignments proceed through an elaborate annotation procedure that comprises a central strategy plus refinement

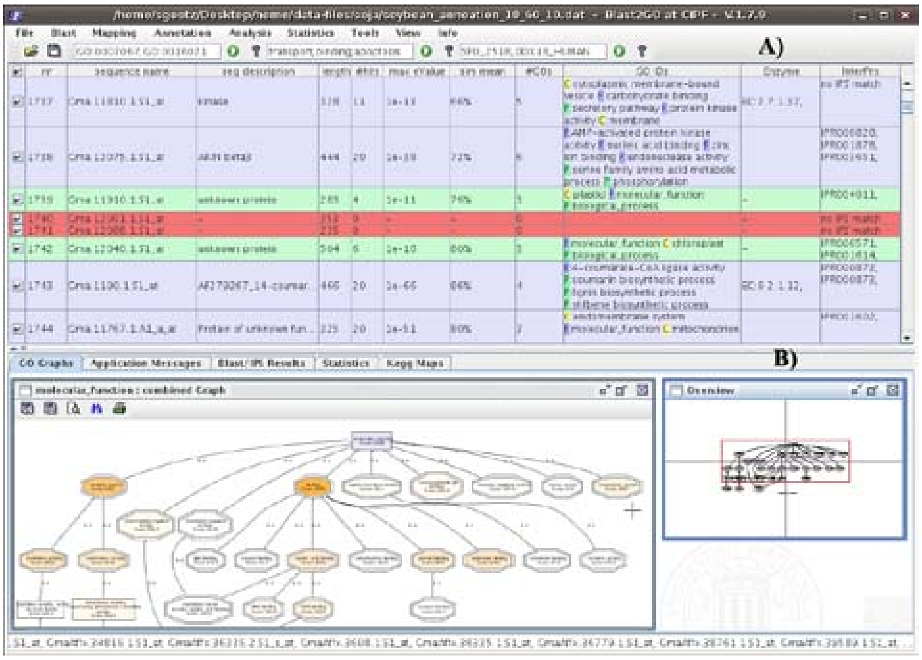

File Blast Mapping Annotation Analysis Statistics Tools View Info
A)
B)
GO Graphs
Application Messages
Blast/IPS Results
Statistics
Kegg Maps
molecular_function : combined Graph
Overview

Figure 2. Blast2GO user interface. A) Main Sequence Menu. Sequences are displayed in a spreadsheet like format. Sequences follow a colour code to indicate the individual analysis status. B) Results tabs showing a Combined Graph. GO terms are coloured by the amount of annotation content.

functions. Next, visualization and data mining engines permit exploiting the annotation results to gain functional knowledge. It is important to stress that Blast2GO is not just a functional annotation method, but a broad and flexible framework to generate and analyse functional information.

The annotation strategy

Once nucleotide or protein sequences are uploaded into B2G, the Main Sequence Menu becomes active and displays sequence data in a spreadsheet like format, which will be incorporating information as generated by the annotation procedure (Figure 2). The amount of sequences Blast2GO admits depends on the user computer capacity but can easily reach several tens of thousands. The Blast2GO annotation procedure consists of three main steps: BLAST to find homologous sequences, *mapping* to collect GO terms associated to multiple BLAST hits and *annotation* to assign trustworthy information to query sequences. Once GO terms have been gathered additional functionalities enable processing and modification of annotation results.

BLAST step. The first step in B2G is to find sequences similar to a query set by BLAST [6]. BLAST can be launched remotely against public databases, such as NCBI nr or Swissprot –default option- or locally when a fasta formatted database and BLAST installation is available at the user site. At this homology search step, the user must further define a BLAST e-value threshold, a minimum value for the length of the matching hsp and a maximal number of retrieved hits. BLAST results are parsed by B2G and displayed through the Single Sequence Menu.

Mapping step. Mapping is the process of retrieving GO terms associated to the hits obtained after a BLAST search. B2G follows diverse mapping strategies from the gene IDs to the GO-database to gather as much annotations as possible from the multiple hit sequences.

Annotation step. In this process, the pool of functional terms collected from each BLAST hit is evaluated to finally assign a functional annotation to the query sequence. Blast2GO integrates different parameters into a single annotation formula to select GO terms. These parameters are: the percentage of sequence similarity, the annotation evidence of the original annotations and the annotation confluence at a GO term from its children terms. Furthermore, exclusion filters can be set on the BLAST e-value score and the percent-

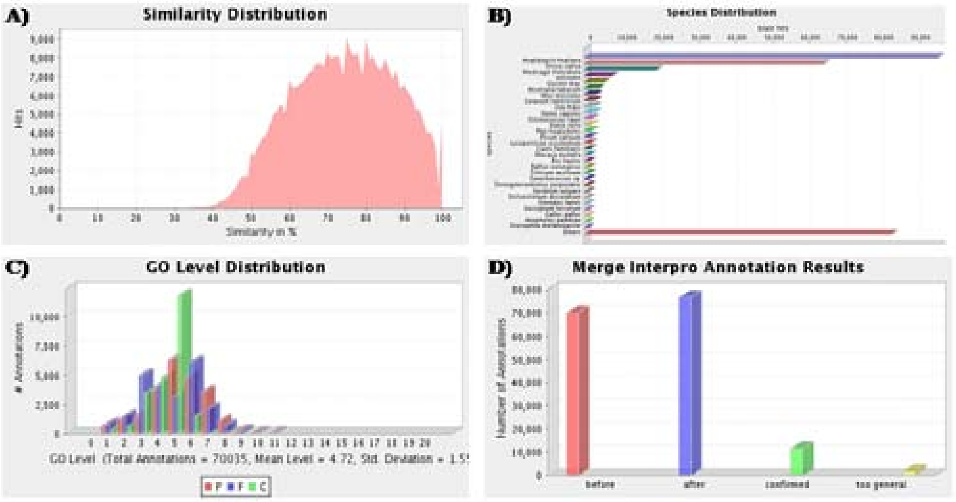


Figure 3. Blast2GO statistical charts. A) e-value distribution for the BLAST hits. B) Species distribution: number of times each species appears in the BLAST results. C) GO level distribution: number of annotations per level for all 3 GO categories. D) InterProScan merging: statistics on the contribution of InterProScan to Blast-based GO annotations.

age of the hit sequence matched by the BLAST alignment. All this results into an annotation score for each GO term and the functional assignment is done by selecting most specific terms above a user defined threshold. Once a GO annotation is available; mapping to Enzyme Codes [7] and KEGG [8] pathways goes automatically.

InterPro. B2G additionally searches for functional information in the InterPro databases [9]. When input data are nucleotide sequences a first translation into protein is done and the longest ORF is selected for querying the EBI InterProScan web service on a total of 13 different motif databases. Results are parsed and integrated into the B2G Main Sequence Table and, if wanted, merged with the GO annotation obtained from the BLAST approach.

Annotation refinements. Three additional functionalities are available for refining annotation results. Manual curation: the user can edit and modify sequence name and GO annotations. Annex augmentation: through this module, biological process and cellular component terms are added from molecular function annotations following the Second Layer scheme [10]. GOSlim projection: the GOSlim is a reduced version of the Gene Ontology which can be used to simplify and condense annotation results. Different GOSlim´s are available, from generic to species specific.

It is worth mentioning that although Blast2GO annotation is clearly high-throughput, sequence information is maintained separately. This implies that each of the above modules is run sequence-wise and therefore can also be modified sequence by sequence. Also the annotation data of each sequence can individually be inspected. A colour code indicates the analysis status of each sequence, from red (unsuccessful BLAST result) to blue (successful annotation), which can be used to select sequences and re-run modules with different parameters. This is useful, for example, to elaborate an annotation strategy that applies sequentially different stringency conditions on the set of sequences.

Graphical analysis

Blast2GO is strongly based on graphical output to provide information. Next to the sequence colour codes, statistical charts and information-rich DAGs can be generated to evaluate analysis results.

Statistical charts. These charts help the user to understand how the annotation proceeded and to decide on the values to give to annotation parameters. Available charts include e-value, similarity and hit species distributions within BLAST results; evidence code and database source of retrieved GO terms, analysis success after each annotation step, average number and depth of the resulting annotation and augmentation statistics for the Annex and InterProScan modules (Figure 3).

Highlighted-DAGs. A core functionality of Blast2GO is the generation of high performance GO direct acyclic graphs (DAGs) through the "Combined Graph" function (Figure 2B). Annotation data can be displayed at each node and highlighting functions are provided to stress the DAG areas where annotation is most concentrated. Additionally, filtering and pruning functions on the node information content are available to generate graphs with only the most relevant information. All this features make the Combined Graph function a powerful module to analyze the collective biological meaning of a set of sequences.

From Blast2GO to Babelomics

The Blast2GO tool was originally designed as a Java desktop application for both annotation and data mining on functional data. Recently, the B2G annotation and graph display modules have been integrated as web services into the Babelomics suite whereby functional profiling can be now more powerfully carried out at Babelomics.

Functional schemes @ Babelomics

Functional profiling methods depend upon the definition of gene lists based on biological properties of interest, whose differential behaviour is analysed. Although the Gene Ontology is the most widely applied vocabulary to functionally relate genes, other information sources can be conceived and are supported at Babelomics. These include alternative functional schemes such as KEGG, InterPro and BioCarta [11]; gene regulation data, such as transcription factor information obtained from Transfac [12] and CisRed [13] or microRNAs from the miRBase database [14]; and disease-related and chemical terms extracted from text-mining technologies [15]. Additionally, tissue and disease specific gene expression profiles are included in Babelomics to serve as profiling comparison sets for user's expression data. All this functional information is supported by Babelomics for humans and partially for other model organisms such as *Caenorhabditis elegans*, *Danio rerio*, *Drosophila melanogaster*, *Mus musculus*, *Rattus norvegicus*, *Saccharomyces cerevisiae*, *Bos taurus* and *Gallus gallus*. The recent integration of Blast2GO makes now available to the Babelomics system the GO, KEGG and InterPro functional annotation for any species.

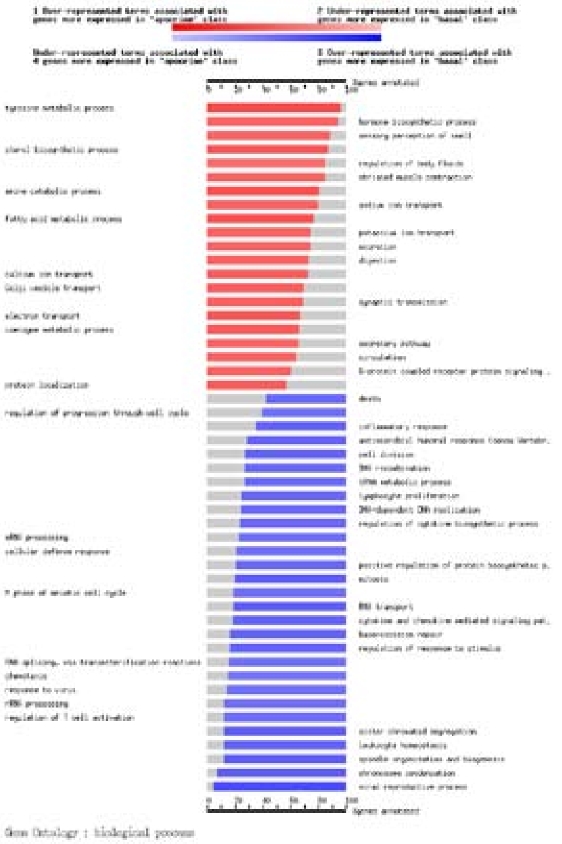


Figure 4. Fatiscan result. Over and under-represented GO terms are found associated to the upper and lower tails of a list of genes ordered by its association with a phenotype.

Testing strategies in functional profiling

FatiGO. FatiGO is the Babelomics implementation of the common enrichment analysis where a contrast is established between a group of sequences of interest and a comparing dataset to identify functional classes which are represented in different proportions. This is of use to interpret the biological meaning of a group of differentially expressed or co-regulated genes identified in functional profiling experiments. FatiGO uses the above described functional classifications and a variety of statistical tests to assess for functional enrichment. Significance values are provided with correction for multiple testing [16].

FatiSca. Babelomics includes a particular adaptation of the so-called Gene Set Enrichment Analysis (GSEA) method: the segmentation test FatiScan [17] associated to genes ranked in a list (Figure 4). GSEA and FatiScan are inspired in systems biology focus on collective properties of gene modules and are free of cut-off thresholds which are the major drawback of the enrichment analysis procedures. FatiScan has the advantage of being independent from both the type of experiment that generated the data and from the experimental design and therefore can be applied to any analyse data from any functional genomics technology. FatiScan analysis has demonstrated to detect functional signatures from experiments where no significant results could be found otherwise [17].

Conclusions

One decade after the start of the omics revolution, it has become clear that the function oriented analysis is the most meaningful and probably only effective way to address the study of these huge data volumes. Many tools have been created along the way to serve to this need. The conjunction in one site of the extensive Blast2GO and Babelomics suites makes for the first time high-throughput functional annotation and advanced functional profiling easily accessible to any organism under study. This is just the beginning, though. As more genetic, interaction and regulation information becomes available, new challenges are posed to data analysis, and new strategies will need to be envisaged to establish the link between the functional organization of the cell and the complexity of the phenotype.

References

[1] Ashburner M, et al. (2000) Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. Nat Genet, 25, 25-29.

[2] Conesa A, et al. 2005. Blast2GO: a universal tool for annotation, visualization and analysis in functional genomics research. Bioinformatics. Sep 15;21(18):3674-6.

[3] Conesa A and Götz S. Blast2GO: A comprehensive suite for functional analysis in plant genomics. The International Journal of Plant Genomics. In press.

[4] Al-Shahrour F et al. (2006) BABELOMICS: a systems biology perspective in the functional annotation of genome-scale experiments. Nucleic Acids Res, 34, W472-476.

[5] Al-Shahrour F, et al. (2005) BABELOMICS: a suite of web tools for functional annotation and analysis of groups of genes in high-throughput experiments. Nucleic Acids Res, 33, W460-464.

[6] Altschul SF, et al. (1990) Basic local alignment search tool. J. Mol. Biol., 215: 403-410.

[7] Bairoch A (2000) The ENZYME database in 2000. Nucleic Acids Res 28:304-305.

[8] Kanehisa M, et al. (2004) The KEGG resource for deciphering the genome. Nucleic Acids Res, 32, D277-280.

[9] Mulder NJ, et al. (2007) New developments in the InterPro database. Nucleic Acids Res, 35, D224-228.

[10] Myhre S, et al. (2006) Additional Gene Ontology structure for improved biological reasoning. Bioinformatics, 22: 2020-2027.

[11] http://www.biocarta.com/

[12] Matys V, et al. (2006) TRANSFAC and its module TRANSCompel: transcriptional gene regulation in eukaryotes. Nucleic Acids Res, 34, D108-110.

[13] Robertson G, et al. (2006) cisRED: a database system for genome-scale computational discovery of regulatory elements. Nucleic Acids Res, 34, D68-73.

[14] Griffiths-Jones D, et al. (2006) miRBase: microRNA sequences, targets and gene nomenclature. Nucleic Acids Res, 34, D140-14

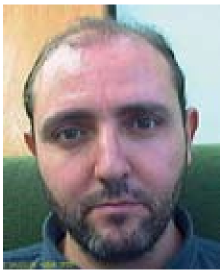
[15] Minguez P, et al. (2007) Functional profiling of microarray experiments using text-mining derived bioentities. Bioinformatics, 23, 3098-3099.

[16] Al-Shahrour F, et al. (2005) Discovering molecular functions significantly related to phenotypes by combining gene expression data and biological information. Bioinformatics, 21, 2988-2993.

[17] Al-Shahrour F, et al. (2007) From genes to functional classes in the study of biological systems. BMC Bioinformatics, 8, 114.

The CIPF Bioinformatics research team.

99 bottles of beer on the GRID



José R. Valverde

EMBnet/CNB, CNB/CSIC,
C/Darwin, 3, Madrid 28049

A monk asked Fuketsu: `Without speaking, without silence, how can you express the truth?' Fuketsu observed: `I always remember springtime in southern China. The birds sing among innumerable kinds of fragrant flowers.'

Ladies and Gentlemen, let me introduce you to the Grid

In the last years clusters have spread and gained popularity as a means to attain greater computing power at lower prices, allowing research groups to increase their familiarity with parallel programming. After the completion of the human genome sequence and the subsequent revolution of genomic and proteomic sciences, our computing needs have exploded beyond the capacities of affordable clusters. Running programs to solve the new problems may require hundreds or thousands of CPUs, something that we may afford "cheaply" by building a larger cluster with "components of the shelf" (COTS clusters) but it still requires a significant investment in money. But then our problem becomes a different one: once we gather these hundreds or thousands of CPUs, where do we put them and how do we maintain the resulting system?

It may be easy to reach an agreement among various groups to make a joint purchase of the machines and share them but finding a suitable "computer room" may easily become terribly expensive (witness *Marenostrum*, the supercomputing cluster in Barcelona, Spain, one of the most powerful systems in the world, built as a cluster of several thousand machines inside an old tem-

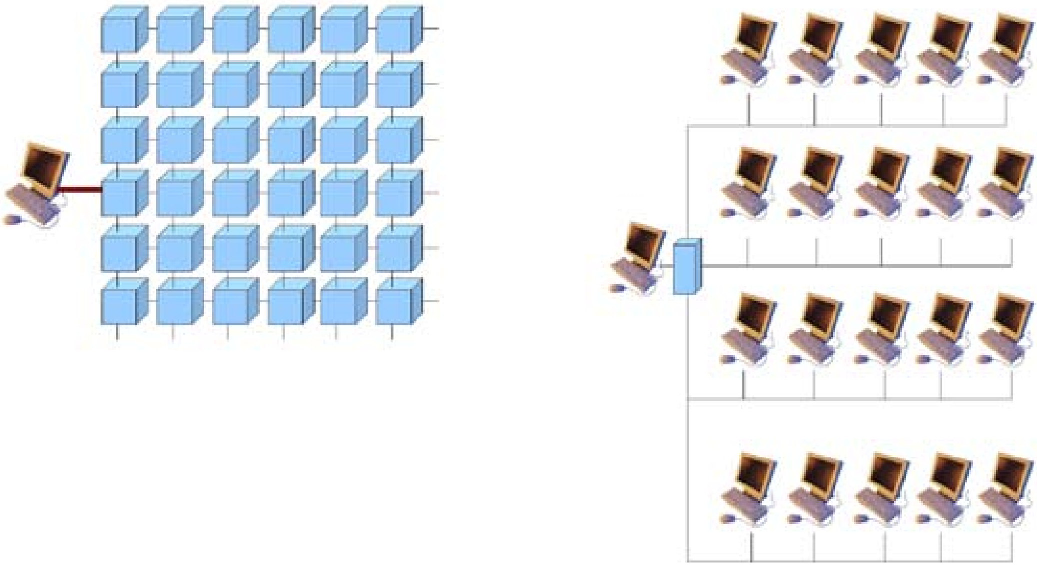


Figure1. A highly parallel supercomputer (e.g. the MasPar of the 1990s) would be built using a large number of CPUs (4-32K) connected among themselves using a highly efficient communications network, and would be accessed from the outside through a specially dedicated system or workstation (e.g. a VAX, or a DEC MIPS workstation). The user would log in to the front-end and do all work from there. The original systems would enclose all components (except the front-end) on a single box. More recent supercomputers (e.g. *Marenostrum*) are custom-built and use independent, high-end workstations deprived of the screen and piled up together in racks and connected with a high-speed network (e.g. Myrinet) instead of bare-bones CPUs, and still uses a front-end to provide user access to the facility. In some respects it is like a glorified supercluster. What makes it special is the sheer number of components requiring special cooling, power, maintenance, security and administration.

ple). With classic clusters we did not have this problem: each group would buy its own small cluster and maintain it locally... The obvious solution would be to attempt something in between both approaches: the clusters are already there, if we could join them together into a single facility, just like a supercomputer, but with machines spread all over the world, we might get the best of both worlds. There is a cost though: we can achieve high speed communications and low latencies over short distances (e. g. on a supercomputing cluster like *Marenostrum*) but not over large distances (like between geographically separated clusters). The end result is a "supercomputing facility" spread all over the World, composed of many clusters of closely coupled machines joined by relatively slow lines.

There are several problems to this approach:

- efficiency: internal communications within clusters happen at high speed, but lines among clusters are relatively slow
- security: internal communications run only over the internal cluster network, but external communications must cross the Internet where they might be intercepted
- trust: the owner of a cluster has full control over his machines, but how can one trust the owners of the machines at other sites?

These and similar problems need to be addressed if we want to build such a shared supercomputing facility, and are the subject of **Grid computing**.

The basic idea of the Grid is simple: we want to gather many machines (individual systems or clusters) spread geographically in a common virtual facility shared cooperatively by all users and used as if it were a single cluster.

Obviously the geographical characteristics impose limits and conditions: we must devise a new way to spread computation over machines with heterogeneous capacities and connection speeds, and we need a software that provides a homogeneous and secure access to these machines (the "middleware", or components that get in the middle of the user and the virtual machine simulated by the Grid).

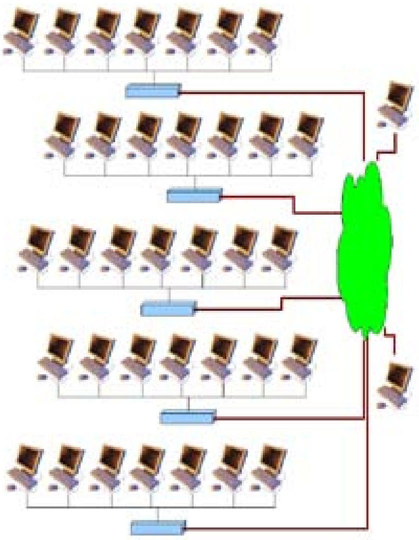


Figure 2. We may build a "virtual supercluster" or "metacluster" by taking advantage of the Internet to join many existing clusters into a new, bigger one. We will need to write new software to deal with the specific properties of this new construct, like distributed data and user management, handling of trust, resource allocation, etc... The end result will be known as a **Grid** and the new software developed will be known as **middleware**. Note that since we are using the Internet to drive communications, we can join any machine, anywhere in the world to the new system, and any means any: as a matter of fact, a system may have several different roles simultaneously, and we may have more than one front-end, worker node, etc.., as many as we need or want.

On the surface a user works not unlike the way one would on a supercomputer of yonder: we will use a special language to describe our jobs, a set of commands to manage job queues and parallel development libraries to build new programs. Before we can make use of the supercomputing resources we must first access a front end node (called "User Interface node", UI on the Grid) and use it to launch our programs to the virtual parallel supercomputer simulated by the Grid.

The main difference is that in this case, and in spite of the classical supercomputer which has a single front-end node, we may have many front-ends (UI nodes), each one belonging to a different person or group. For this reason we need to identify ourselves twice: one to access the UI (which may belong to anybody, possibly untrustworthy) and a second to access the Grid. The identity on the UI will be provided by its owner

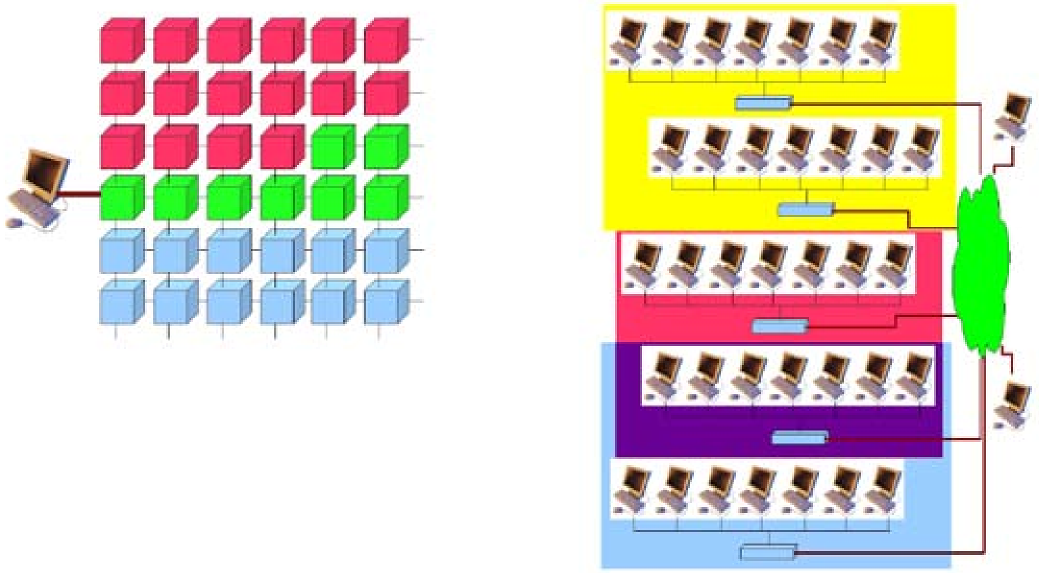


Figure 3. A very common problem on classical parallel supercomputers happened when a community needed to run a program that would not require all the CPUs available: just running this program would not make full use of a very expensive system. The usual solution would be to partition the machine into virtual subsets (also called partitions or CPU sets), and devote each subset to solve a different problem; then various user communities would be able to run their large problems simultaneously. We can do the same on the Grid: we can divide the CPUs available into subsets, each devoted to solve the problems of a specific community, calling each set a virtual organization (VO). The difference now is that since this is not a single system with a single owner, each cluster owner has freedom to decide which communities (or virtual organizations) will be able to run their programs on his cluster of machines, i. e. a single CPU or cluster may support more than one VO.

(ourselves if we install the UI software in our computer) and the identity of the Grid will be provided by a special manager designed by the Grid administrators. In practice, we will have to get an "identity certificate" from a "certification authority" (CA) that is trusted both by us and the Grid designed managers, and these will simply grant access to the Grid for our certificate. This means that all trust relies on the CA (which is by definition trustworthy) and on ourselves. hence there is no need to trust neither Grid administrators, nor managers, nor owners of UI nodes.

In short, to work on the Grid we need

- the first time
  - ° contact with a mutually trustworthy (to us and the Grid administrators) CA to get an identification certificate
  - ° get the manager of a VO (more on this later) to grant access to the Grid for our certificate
  - ° get an account on a UI node connected to the Grid which we can use to submit jobs using our certificate
- whenever we want to work
  - ° connect to the UI node using the username and password we were assigned by the UI manager
  - ° identify ourselves to the Grid using our certificate
  - ° submit to the Grid any jobs we want to execute using
    - * a job description (using JDL, the Job Description Language)
    - * the software to be run
    - * the data we want to process
  - ° recover our results from the Grid

As we can see, for a user, working with the Grid is not very different from the way we have been working with other systems in previous decades using batch queues. Below the surface, instead of a parallel computer we have a Grid behaving as if it were one, although in reality it consists of thousands of computers spread geographically and belonging to different owners.

What about this VO thing we mentioned? Actually the Grid is a general concept that can be used for many purposes. Conversely, the middleware may be used by many communities, and a member may be willing to participate in more than one community. As a result what we do is build a generic Grid infrastructure joining all the computers from everybody willing to use this technology, and then "partition" it into subsets, where each subset corresponds to a community of users and is composed by those machines whose owners want to share them with all other members of that community. Each of these subsets is called a "virtual organization" or VO for short.

EGEE: bringing supercomputing to the masses

If you understand, things are such as they are; if you don't understand, things are such as they are.
(Gesha)

In 2003 the project **EGEE** (**Enabling Grids for E-science in Europe**) gets the green light from the EU and gets started in April 2004. EGEE has been designed to provide "the" production Grid infrastructure for Europe and uses a special middleware named **gLite**. gLite is composed by standard components approved by the **Global Grid Forum, GGF** (now called **Open Grid Forum, OGF**) plus a number of extensions that make developer and user lives a lot easier. gLite providing additional facilities has become popular in other infrastructures outside EU, like ChinaGrid, EUMedGrid, EELA... As EGEE has been expanding beyond Europe, the acronym has evolved to mean *Enabling Grid for E-sciencE*.

EGEE provides access to a Grid spanning many countries, both inside and outside Europe, and connects thousands of computers. Software development and Grid access is done using specially designed computers known as "**User Interface nodes**".

Thus, the first thing we need is a user name and a password to access a UI node. However, since anybody can install a UI if desired (you too), and the UI belongs to its owner (obviously), this is not enough to grant access to the vast resources provided by the Grid: we want to know who does what with the Grid, and for this we need to be sure of the identity of each and every user. Therefore, to access the Grid from a UI we must identify ourselves using a **personal certificate** issued by a **Certificate Authority** accepted by the Grid administrators. This certificate must be stored in the UI, and since we don't want to trust the UI managers, we usually store it protected by a *pass phrase* (something much longer than a password to be safer).

In other words it is like accessing an automatic teller machine in a bank: it is not enough that they open the door to the office for us (with our username and password), we also need a valid credit card (our Grid certificate) issued by an entity (the CA) that is accepted by the bank owners (the Grid), and that is protected by a password known only to us,

Now, we are going to make the Grid sing '99 bottles of beer'. For this we need:

- Contact a UI computer
  - using our username and password

- Contact with the Grid using our certificate
  - `voms-proxy-init`
  - enter the password we use to protect our certificate

- Prepare our job
  - prepare the program (see Fig.4)
  - prepare the data (you won't need any data for this job)
  - prepare a description of the program and data using JDL (see Fig. 5)

- Submit the job
  - `edg-job-submit 99bob.jdl`
  - will produce a URL that identifies our job on the Grid

- Monitor our job status using the produced URL
  - `edg-job-status http://xxxxx/xxxxxx`

- Once the job is finished, the results will be stored on the Grid and we'll need to recover them
  - `edg-job-get-output -dir . http://xxxxx/xxxxxx`

- Disconnect from the Grid

  - `grid-proxy-destroy`
- Exit from the UI
  - `logout`

We should note some relevant details in this process:

- to begin with, the default way to work with the Grid is by submitting jobs in batches (with 'edg-job-submit')
- job properties must be described using a special language (JDL) in a separate file
- we can submit parallel jobs (just say so in the JDL)
- job output is usually generated in a global temporary directory, but we can request it be saved anywhere (here we ask for a subdirectory of our current "--dir ." directory)
- to really access the Grid we need the certificate: we should protect it carefully with a pass phrase
- we specify the VO to use as we may belong into more than one (in this case "biomed")
- we never trust anybody, nor the UI not its administrators: Grid access is personal and untransferrable
- when we connect to the Grid we must state for how long we will be working (how long will our work session last) and we can renew our session at any time.

```
#!/bin/bash
# Bourne Again shell version of 99
Bottles
# Dave Plonka - plonka@carroll1.cc.edu

typeset -i n=99
typeset bottles=bottles
typeset no

while [ 0 != $[ n ] ]
do
   echo "${n?} ${bottles?} of beer on
the wall,"
   echo "${n?} ${bottles?} of beer,"
   echo "take one down, pass it
around,"
   n=n-1
   case ${n?} in
   0)
      no=no
      bottles=${bottles%s}s
      ;;
   1)
      bottles=${bottles%s}
      ;;
   esac
   echo "${no:-${n}} ${bottles?} of
beer on the wall."
   echo
done

exit
```

Figure 4. a shell script to sing 99 bottles of beer. Taken from the **99 bottles of beer web** site (`http://www.99-bottles-of-beer.net/`). Copy this program into a file and save it as '`99bob.sh`'.

```
Type    = "job";
JobType = "normal";
VirtualOrganisation = "biomed";
Executable = "99bob.sh";
StdOutput = "where";
StdError = "horror";
OutputSandbox = { "where" };
```

Figure 5. JDL instructions to execute the 99bob shell script.

In general, and as can be seen, accessing the Grid is not very different from using any other batch system, but it looks strange for those who are familiar only with interactive computing. Certainly, there is still plenty of room to improve the way we use the Grid, and this is an active field for development.

In future installments we will see in more detail how Grid technologies can be applied to solve real Bioinformatics problems, starting with additional details on job submission and management which we will review in the next article in the series.

Acknowledgements

We want to thank EMBnet[1] for making publicly available its education web site [2], and the EU for its support to projects EGEE[3] (INFSO-RI-031688) and EMBRACE[4] (LHSG-CT-2004-512092) which have allowed us to do this work.

References

[1] `http://www.embnet.org`

[2] `http://edu.embnet.org`

[3] `http://www.eu-egee.org`

[4] `http://www.embracegrid.org`

The hands to say it

Vivienne Baillie Gerritsen

When I was a little girl, I thought that my left-handed classmates were special. I envied their difference. And I used to marvel at the way they crouched over their desk, embracing something invisible as they did their best to avoid smudging ink all over their sheet of paper. Left-handedness *is* special. But so is right-handedness. Humans are not the only animals to make use of their hands – or claws, or paws, or hooves - but they are the only ones who show a marked preference for either the left one, or the right one. If this is so, there must be a reason for it. And not only must there be a reason but it must translate a certain structure of our brain: an asymmetry somewhere. Indeed, our brain is divided into two hemispheres which are dedicated to processing different activities. One side looks after our dreams, while the other is far more down to earth. LRRTM1 is the first protein to have been discovered which seems to be directly involved in this brain asymmetry. Consequently, it influences the handedness of a human-being and, more astonishingly, may also predispose individuals to psychotic troubles such as schizophrenia.

Two men engaged in conversation

Source unknown

Humans are particularly clever with their hands. One of the very first special events in our evolution was to get onto our hind limbs and free our hands for collecting food and making tools. A subsequent good move was to take away the burden of communication on our hands by developing our vocal instruments. Indeed speech, and its fine-tuned elaboration that only humans have managed to master so far, has given our hands great freedom which we have put to use in a multitude of ways. But none of this can explain why we are – for the great majority (90%) – right-handed. Hosts of other species also use their appendages for collecting food, eating or grooming but they don't have a distinct preference for one hand over the other.

The passing of roles from hand to mind expresses a particular brain structure. In turn, the progressive use of speech has continued to mould our brain into a shape peculiar to the human species. But why would that make us right-handed? For speech to evolve, one part of our brain had to evolve differently too, and in so doing it made most humans right-handers. This is what is known as the 'right-shift factor'. Consequently, our right-handedness is not the result of nature selecting right-handers over left-handers but rather of nature nudging our brain into a shape which encourages the act of speaking. As a result, over the millions of years, the human brain has been divided into two hemispheres. The right hemisphere is dedicated to the world of emotions and imagination, whilst the left hemisphere deals with talking and logical processes.

LRRTM1, or leucine-rich repeat transmembrane neuronal protein 1, is a protein involved in brain development. It has a set of repetitive domains in its sequence which are known to be involved in protein-protein interactions, a vital activity in the light of brain structure and development. LRRTM1 may be one of the factors which bestow upon the brain its asymmetry. It is expressed very early in the development of forebrain structures and may function in neuronal differentiation and connectivity. It is also thought that it could have a role in intracellular trafficking in axons. Left-handedness, which is handed down by the father, may well be due to LRRTM1 dysfunction causing the original asymmetry to be flipped around, or reduced. However, it has been pointed out that chimpanzee LRRTM1 is 100% identical to human LRRTM1, yet no one has found a left-handed chimp, or an articulate one for that matter. Handedness and subtle brain asymmetry, as found in humans, are the result of much more than just one protein – and environmental factors are undoubtedly of great influence too.

With a role in brain development and possible neuronal connectivity, it is hardly surprising that LRRTM1 has been linked to neuronal diseases such as schizophrenia, autism and language impairment. Likewise, a logical step was to wonder whether left-handedness could not be taken as an indication to a predisposition for neuropsychiatric disorders. It so happens that in one study carried out on schizophrenic individuals many were left-handed. This kind of result has to be taken with caution though. It does not mean that every left-handed individual is prone to some form of psychosis. Many right-handers suffer from psychiatric impairment too. However, it does suggest that genetic components involved in the structure of our brain may be indicative of a predisposition to a neuronal illness, given the environment. Surprisingly, other studies have shown that left-handers are more prone to accidents than right-handers. No clear explanation has yet been given but it may just be because our society is really built for right-handers.

LRRTM1 is predicted to link to another protein – or proteins – where the bond would supposedly trigger off a reaction. With this in mind, if it can be shown that LRRTM1 does have a role in the development of neuropsychiatric diseases, it may well prove to be precious in the design of novel therapies to lessen such disorders. Once again though, no protein acts on its own. There are genes upstream and downstream of LRRTM1 involved in its expression. Furthermore, an individual's environment is hugely important in triggering off a psychiatric disorder: drugs, alcohol, abuse, violence, stress etc. And, besides psychiatric disorders, what to think of someone who writes with their left hand and throws with their right? What is their brain structure? Is LRRTM1 also part of semi left-handedness? It is all very mysterious. But the fascinating part of the story is to realise that were it not for our words, we would not be able to carry out nearly as much as we do with our hands.

Cross-references to Swiss-Prot

LRRTM1, *Homo sapiens* (Human) : Q96DN1

References

1. Francks C., Maegawa S., Lauren J., Abrahams B.S., Velayos-Baeza A., Medland S.E., Colella S., Groszer M., McAuley E.Z., Caffrey T.M., Timmusk T., Pruunsild P., Koppel I., Lind P.A., Matsumoto-Itaba N., Nicod J., Xiong L., Joober R., Enard W., Krinsky B., Nanba E., Richardson A.J., Riley B.P., Martin N.G., Strittmatter S.M., Moeller H.-J., Rujescu D., St Clair D., Muglia P., Roos J.L., Fisher S.E., Wade-Martins R., Rouleau G.A., Stein J.F., Karayiorgou M., Geschwind D.H., Ragoussis J., Kendler K.S., Airaksinen M.S., Oshimura M., De Lisi L.E., Monaco A.P.
   LRRTM1 on chromosome 2p12 is a maternally suppressed gene that is associated paternally with handedness and schizophrenia
   Molecular Psychiatry 12:1129-1139(2007)
   PMID: 17667961

2. Harrison R.M., Nystrom P.
   Handedness in captive bonobos (*Pan paniscus*)
   Folia Primatologica 79:253-268(2008)
   PMID: 18212503

3. Wolman D.
   The secrets of human handedness
   The New Scientist Magazine, Issue 2524, November 5th 2005

National Nodes

Argentina
IBBM, Facultad de Cs. Exactas, Universidad Nacional de La Plata

Australia
RMC Gunn Building B19, University of Sydney, Sydney

Austria
Vienna Bio Center, University of Vienna, Vienna

Belgium
BEN ULB Campus Plaine CP 257, Brussels

Brazil
Lab. Nacional de Computação Científica, Lab. de Bioinformática, Petrópolis, Rio de Janeiro

Chile
Centre for Biochemical Engineering and Biotechnology (CIByB). University of Chile, Santiago

China
Centre of Bioinformatics, Peking University, Beijing

Colombia
Instituto de Biotecnología, Universidad Nacional de Colombia, Edificio Manuel Ancizar, Bogota

Costa Rica
University of Costa Rica (UCR), School of Medicine, Department of Pharmacology and ClinicToxicology, San Jose

Cuba
Centro de Ingeniería Genética y Biotecnolgía, La Habana

Finland
CSC, Espoo

France
ReNaBi, French bioinformatics platforms network

Greece
Biomedical Research Foundation of the Academy of Athens, Athens

Hungary
Agricultural Biotechnology Center, Godollo

India
Centre for DNA Fingerprinting and Diagnostics (CDFD), Hyderabad

Israel
Weizmann Institute of Science, Department of Biological Services, Rehovot

Italy
CNR - Institute for Biomedical Technologies, Bioinformatics and Genomic Group, Bari

Mexico
Nodo Nacional EMBnet, Centro de Investigación sobre Fijación de Nitrógeno, Cuernavaca, Morelos

The Netherlands
Dept. of Genome Informatics, Wageningen UR

Norway
The Norwegian EMBnet Node, The Biotechnology Centre of Oslo

Pakistan
COMSATS Institute of Information Technology, Chak Shahzaad, Islamabad

Poland
Institute of Biochemistry and Biophysics, Polish Academy of Sciences, Warszawa

Portugal
Instituto Gulbenkian de Ciencia, Unidade de Bioinformatica, Oeiras

Russia
Biocomputing Group, Belozersky Institute, Moscow

Slovakia
Institute of Molecular Biology, Slovak Academy of Science, Bratislava

South Africa
SANBI, University of the Western Cape, Bellville

Spain
EMBnet/CNB, Centro Nacional de Biotecnología, Madrid

Sri Lanka
Institute of Biochemistry, Molecular Biology and Biotechnology, University of Colombo, Colombo

Sweden
Uppsala Biomedical Centre, Computing Department, Uppsala

Switzerland
Swiss Institute of Bioinformatics, Lausanne

Specialist Nodes

EBI
EBI Embl Outstation, Hinxton, Cambridge, UK

ETI
Amsterdam, The Netherlands

ICGEB
International Centre for Genetic Engineering and Biotechnology, Trieste, Italy

IHCP
Institute of Health and Consumer Protection, Ispra. Italy

ILRI/BECA
International Livestock Research Institute, Nairobi, Kenya

LION Bioscience
LION Bioscience AG, Heidelberg, Germany

MIPS
Muenchen, Germany

UMBER
School of Biological Sciences, The University of Manchester,, UK

for more information visit our Web site

`www.embnet.org`

EMBnet.news
ISSN 1023-4144

Dear reader,

If you have any comments or suggestions regarding this newsletter we would be very glad to hear from you. If you have a tip you feel we can print then please let us know. Please send your contributions to one of the editors. You may also submit material by e-mail.

Past issues of EMBnet.news are available as PostScript or PDF files. You can get them from the EMBnet organization Web site:
`http://www.embnet.org/download/embnetnews`

Publisher:
EMBnet Executive Board
c/o Erik Bongcam-Rudloff
Uppsala Biomedical Centre
The Linnaeus Centre for Bioinformatics, SLU/UU
Box 570 S-751 23 Uppsala, Sweden
Email: erik.bongcam@bmc.uu.se
Tel: +46-18-4716696

Submission deadline for the next issue:
May 20, 2008