

Data and text mining

Visualizing information across multidimensional post-genomic structured and textual databases

Ying Tao, Carol Friedman[†] and Yves A. Lussier^{*,†}

Department of Biomedical Informatics, Columbia University, 622 West 168th Street, Vanderbilt Clinic, 5th Floor, New York, NY 10032, USA

Received on August 28, 2004; revised on November 5, 2004; accepted on December 6, 2004

Advance Access publication December 14, 2004

ABSTRACT

Motivation: Visualizing relationships among biological information to facilitate understanding is crucial to biological research during the post-genomic era. Although different systems have been developed to view gene–phenotype relationships for specific databases, very few have been designed specifically as a general flexible tool for visualizing multidimensional genotypic and phenotypic information together. Our goal is to develop a method for visualizing multidimensional genotypic and phenotypic information and a model that unifies different biological databases in order to present the integrated knowledge using a uniform interface.

Results: We developed a novel, flexible and generalizable visualization tool, called PhenoGenesviewer (PGviewer), which in this paper was used to display gene–phenotype relationships from a human-curated database (OMIM) and from an automatic method using a Natural Language Processing tool called BioMedLEE. Data obtained from multiple databases were first integrated into a uniform structure and then organized by PGviewer. PGviewer provides a flexible query interface that allows dynamic selection and ordering of any desired dimension in the databases. Based on users' queries, results can be visualized using hierarchical expandable trees that present views specified by users according to their research interests. We believe that this method, which allows users to dynamically organize and visualize multiple dimensions, is a potentially powerful and promising tool that should substantially facilitate biological research.

Availability: PhenogenesViewer as well as its support and tutorial are available at <http://www.dbmi.columbia.edu/pgviewer/>

Contact: Lussier@dbmi.columbia.edu

INTRODUCTION

Visualizing relationships among biological information to facilitate understanding is crucial to biological research during the post-genomic era, in which the volume and complexity of available biological information is increasing at an accelerating rate. Although visualizing molecular networks is intensely pursued by the community, visualizing gene–phenotype relationships, the phenotype (Freimer and Sabatti, 2003), is of equal importance, especially for the approach of systems biology (Tao *et al.*, 2004). Although some systems have been developed to view gene–phenotype relationships for

specific databases, to our knowledge, very few have been designed specifically to meet the requirements for a general flexible tool for visualizing multidimensional genotypic and phenotypic information together. A general tool of information visualization over multiple databases is needed in the postgenomic era and should include the following basic requirements:

- (1) It should be capable of dealing with a large number of dimensions. Related genotypic and phenotypic information as well as contextual information constitute multidimensional datasets, such as DNA sequence, gene, protein, cytogenetic band, chromosome, inheritance mode, phenotype name, organism, assay and bibliographic information. Additionally, phenotypes are compositional and could comprise different phenotypic components (Mahner and Kary, 1997; Freimer and Sabatti, 2003). For example, the phenotype *asthma* in the level of disease diagnosis could have a body location component *respiration system* and a serum test component *elevated serum immunoglobulin E*. These phenotypic components could also be regarded as individual dimensions. Thus, the total number of dimensions could be quite large.
- (2) It should allow flexible queries. To meet the requirement of different users and various purposes, systems should allow users to select dimensions of interest and apply customized filters. For example, a user may want to know the chromosome locus, protein structure and associated phenotypes related to a specific gene. A system should provide the user with the function to define his/her query easily without having to know the underlying database structure or query language.
- (3) It should provide visualization of associative relations based on users' queries so that relational patterns can be easily perceived. For example, a user may want to use a disease-centric view to see all the genes clustered under each of the different types of cancers so that hotspot genes could be found for cancers, and to explore how different types of cancers differ in etiology. Another user may want to obtain a gene-centric view to see all the diseases clustered under a specific gene or a group of genes so that the user could determine the major function of that gene or gene group. Owing to the highly multidimensional nature of genotypic and phenotypic information, a well-organized output presentation could disclose clusters and patterns otherwise difficult to discover.

*To whom correspondence should be addressed.

[†]The authors wish it to be known that, in their opinion, the last two authors should be regarded as joint First Authors.

- (4) It should be able to visualize data integrated from different databases regardless of the communities that develop them. Biological knowledge is currently distributed across multiple heterogeneous databases, which have different focuses and different ways of information organizations. For example, OMIM (Hamosh *et al.*, 2002) is both gene-centric and disorder-centric. Swiss-Prot is protein-centric (Bairoch and Apweiler, 1996). GenBank (Benson *et al.*, 2000) is sequence-centric and genes are regarded as special segments within DNA sequences. Molecular Modeling DataBase (MMDB) (Marchler-Bauer *et al.*, 1999), the structure database of the National Center for Biotechnology Information (NCBI) (Wheeler *et al.*, 2000), is structure-centric. MEDLINE is bibliography-centric (PubMed MEDLINE, <http://www.ncbi.nlm.nih.gov/PubMed/>). A user may want to find all the genes related to a disease from OMIM and then find all their encoded proteins from a protein database. For some proteins a user may need to investigate their sequences, three-dimensional structures and the original papers. Such a process needs information across all the mentioned databases. A visualization system should be able to visualize this information across databases gracefully, although it is not necessary for a visualization system itself to contain the ability of interfacing and integrating multiple databases.
- (5) It should have easy-to-use and efficient user interfaces so that a broad range of biologists without much computer background could learn and use it with minimum training.

In this paper, we present a general visualization tool, called PGviewer, which meets the five basic requirements mentioned previously. Our aim is to develop a general method for visualizing multidimensional genotypic and phenotypic information, and a model to unify interfaces of different databases. Our method uses a tree structure to visualize the clustering relationships of the multidimensional biological information across multiple databases according to users' queries. We demonstrate its flexibility and generalizability over two sets of data.

In the rest of this paper, we will first review existing approaches for browsing, querying and visualizing biological data. Then, we will discuss the details of our system's components, interfaces, algorithms and our evaluation process. Next, results from the evaluation will be given. Finally, we will discuss the advantages and limitations of our methods and future work.

Related work

PGviewer is based on our previous work on (1) organizing phenotypes across genomic databases and (2) visualizing clinical phenotypes. The former methods infer relationships across heterogeneous phenotypes in distinct databases using structured ontologies or computational terminologies (Cantor and Lussier, 2003, 2004; Lussier and Li, 2004). The latter method consists of another tree viewer called DynTreeViewer, which was designed to flexibly display associative relationships between the components of clinical terms obtained from narrative text (Liu and Friedman, 2000; Friedman *et al.*, 2003). For example, it could display a problem-oriented view of clinical terms occurring in patient reports or a body location-oriented view. Its tree organization is similar to that of PGviewer. However, PGviewer is more flexible than DynTreeViewer. In DynTreeViewer,

to modify a tree view users can change the clustering order only by bringing a level or dimension of a tree to the top level of that tree. Users cannot specify the order of dimensions below the first level. PGviewer provides full flexibility by allowing permutation of dimensions' ordering in all levels of a tree. Another difference is that PGviewer uses a relational database to manage data instead of native XML in order to improve efficiency and scalability and to take advantage of standard database query functions.

The implementation concept of PGviewer is from the n -dimensional data cube (Gray *et al.*, 1996), an established method for organizing multi-dimensional databases, and an important interface for data cube, Pivot Table (Graefe *et al.*, 1998). The Pivot Table allows the data cube to be rotated or pivoted, so that different dimensions of the dataset can be arranged into a two-dimensional table. PGviewer inherits the Pivot Table's feature of flexible data definition. PGviewer differs from the Pivot Table in that it displays results using a hierarchical expandable tree instead of using tabular results. Another difference is that the Pivot Table is more suitable for analysis of numeric values but PGviewer is designed to show associated relationships of nominal data.

There are currently a number of systems aimed at browsing, querying and visualizing biological entities and their relationships. The differences between our system and these existing systems are summarized as follows:

Pre-defined visualization. This group of systems returns output in pre-defined views according to users' searching criteria. Searching results are formatted in pre-defined tables. Obtaining information across different databases is implemented by the hyperlinks embedded in searching results. Actually, this approach is taken by most of the databases, such as NCBI (Wheeler *et al.*, 2000), Mouse Genome Informatics (MGI) (Bult *et al.*, 2004), Flybase (FlyBase Consortium, 2003) and GeneCards (Rebhan *et al.*, 1998). Technically, this browsing approach is very flexible and can be extended to any number of dimensions just by selecting available hyperlinks. Different databases are easily coordinated by URL links. However, the disadvantage is that the search interfaces focus on one fixed dimension and the returned information is organized according to a predefined view. Users are required to integrate the related information manually by selecting all the hyperlinks laboriously when they need to retrieve related information. The associative relationships of objects across multiple databases are not easily seen. This process is likely to be inefficient owing to excessive number of branches to obtain the complete data. Our system differs from these systems because it allows users to define their information needs in one step without multi-screen browsing. Furthermore, relationships among dimensions from different databases are visualized in a tree structure within the same view so that patterns are easily perceived.

User-defined queries. This group of systems attempt to avoid the disadvantages of pre-defined visualization by a centralized platform and allow flexible queries using special querying languages, such as TAMBIS (Baker *et al.*, 1998), Kleisli (Wong, 2000) and TINet (Eckman *et al.*, 2001) or using query generation interfaces (Chen *et al.*, 1998; Kasprzyk *et al.*, 2004). Owing to the flexibility of query scripts and query generation interfaces, in this approach a user can freely define informational dimensions in queries across different databases. Thus, this group of systems meets the requirement for dealing with a large number of dimensions, allowing flexible queries and coordinating heterogeneous databases. However, they

concentrate on flexible queries but not on flexibility in visualizing the resulting relationships of the biological entities because most of them use flat tables as the format of the query result. Therefore, when a result is large, associative relationships are hard to discover within a large table. In addition, in the approach of using special querying languages, the requirement for understanding special syntaxes as well as database schemas may affect its broad use. Our method maintains the feature of using a graphical query generation interface to generate flexible queries. The major difference from these systems is that our system visualizes retrieved results in an organized manner in order to facilitate better understanding.

Other systems. Graphic visualization systems for molecular network have been investigated extensively (Kolpakov *et al.*, 1998; Koike and Rzhetsky, 2000; Jenssen *et al.*, 2001; Karp, 2001) but they are not designed to visualize gene–phenotype relationships. A few systems do display the relationships of genes and phenotypes graphically, e.g. SemGen (Rindflesch *et al.*, 2003) and g2p (Bodenreider and Mitchell, 2003). However, these systems visualize only two dimensions of entities, namely, genes and phenotypes and no other related information. There is a general tool, called BITOLA, for exploring user-specified classes of bio-medical terms from MEDLINE in a large scale (Hristovski *et al.*, 2003). It is flexible in that users can specify the classes of dimensions they are interested in. However, the input is focused on one database and the output is tabular by design and no more than three dimensions can be displayed at one time. There is another group of graphic tools for visualizing Gene Ontology (GO) (Ashburner *et al.*, 2000) annotation information based on a large number of input genes (Zeeberg *et al.*, 2003; Zhong *et al.*, 2003; Al-Shahrour *et al.*, 2004; Zhang *et al.*, 2004). These tools concentrate on visualizing collective profiles of phenotypic annotation based on a group of genes rather than individual genes. The purpose is different from the one discussed in this paper.

SYSTEM AND METHODS

System components

The proposed visualization methods (PGviewer) are described below.

The basic idea behind our system is the following: databases contain objects and objects are described by attributes. All attributes within all the objects in all the databases constitute the dimensions in the whole data space. Users' queries can be formed by selecting an ordering of these dimensions with filtering criteria on each dimension. To be presented to users, the results of a query are arranged in a tree structure so that users can explore the result space clustered through associative relationships according to their needs. It is important to note that the tree structure we use represents an ordering or clustering of information, and should not be associated with a hierarchical classification, which is a typical use of a tree when specifying an ontology or taxonomy. Based on these methods, the PGviewer user interface consists of two parts, namely (1) a query definition interface and (2) a presentation interface of the query result.

The architecture overview of our system is illustrated in Figure 1.

Denormalized database

PGviewer operates over a denormalized database (Fig. 1). In order to generate this denormalized database, we integrate in a semi-automated way independent databases using PERL scripts, cross-indexes and 'SQL join' commands. We then denormalize the relevant fields. Two datasets (human genomics, mouse genomics) are used to demonstrate that our method is generalizable. The human genomics dataset shows gene–phenotype relationships collected in OMIM, which were obtained by human manual curation. The mouse genomics dataset shows gene–phenotype relationships extracted from a subset

of MEDLINE related to the mouse model organism. This collection consists of information extracted from Medline citations using a revised version of a natural language processing (NLP) extraction and encoding system called BioMedLEE (Chen and Friedman, 2004). BioMedLEE was developed based on components of two established NLP systems, the components of MedLEE (Friedman *et al.*, 1994) enhanced with a small number of additional grammar components from GENIES (Friedman *et al.*, 2001; Krauthammer *et al.*, 2002). MedLEE has been used operationally in the clinical domain to encode information in textual patient reports since 1995 and has been shown to actually improve patient care. GENIES, which is an adaptation of MedLEE, extracts biomolecular interactions from the literature. It is a component of the GeneWays system (Rzhetsky *et al.*, 2000, 2004), and has been used to process over 100 000 full journal articles, in order to populate the GeneWays knowledge base.

- (1) The human genomics dataset was obtained from the entire OMIM Gene Map table downloaded from the OMIM website, which contains 9042 entries of gene–disorder relationships. For this dataset, we extracted gene name, gene location, disorder and OMIM ID from this table. We also obtained the bibliographic information for each OMIM entry using a script to read OMIM's website. To disclose the molecular mechanism of human hereditary diseases, we added GO terms for each OMIM entry via LocusLink (Maglott *et al.*, 2000). The files we used are mim2loc and loc2go downloaded from the OMIM website. We have nine dimensions in our human genomics dataset: (1) OMIM_ID (including OMIM title) (2) gene location (3) gene (4) GO_term (5) disorder (6) PubMed_ID (including article titles) (7) year (8) journal and (9) authors.
- (2) The mouse genomics dataset comes from three databases: (1) a subset of MEDLINE citations related to the mouse model organism; (2) gene and phenotype relationships extracted from these articles using BioMedLEE, where the phenotypes are encoded using identifiers of the Unified Medical Language System (UMLS); and (3) a UMLS–GO mapping database (Sarkar *et al.*, 2003) that map terms from UMLS (Lindberg, 1990) to GO terms.
 - (a) MEDLINE citation information. We collected bibliographic information, including PubMed ID, article title, journal, publication year and authors, from the MEDLINE subset. There are over 1200 citations in this subset. As the original files from MEDLINE were in the XML format, an XML parser written in PERL was used to flatten the files before they are imported into our database.
 - (b) MEDLINE articles parsed by BioMedLEE. Genotypic and phenotypic information were extracted from the titles and abstracts of the MEDLINE subset. BioMedLEE was used to process the titles and abstracts and to extract the relevant information. Extracted information includes gene names, phenotypes and phenotype-related biological structures. BioMedLEE can encode phenotype and biology structure into various terminology codes, and in this particular paper we used UMLS codes. The output is in the structured format of XML. A simplified version of the output from BioMedLEE is shown below for the sentence from a MEDLINE abstract 'Tsc2 heterozygote display 100% incidence of multiple bilateral renal cystadenomas, 50% incidence of liver hemangiomas, and 32% incidence of lung adenomas by 15 months of age'. The tag represents the type of information whereas the attribute **v** represents the value. Note that, the value for gene tags display the full form of the gene. The outermost tags represent the primary type of information (e.g. **gene**, **phenotype**); nested tags represent modifiers of that information (e.g. **genemod**, **anatomy** and **region**). The tag **phenotype** is a semantic type associated with diseases and other abnormalities. The tag **sid** is a tag identifying a sentence. For example, the last **phenotype** tag in the example below has the value 'adenoma', which is modified by a body organ 'lung', measurement information '32%' and a sentence ID 's1.1.1':

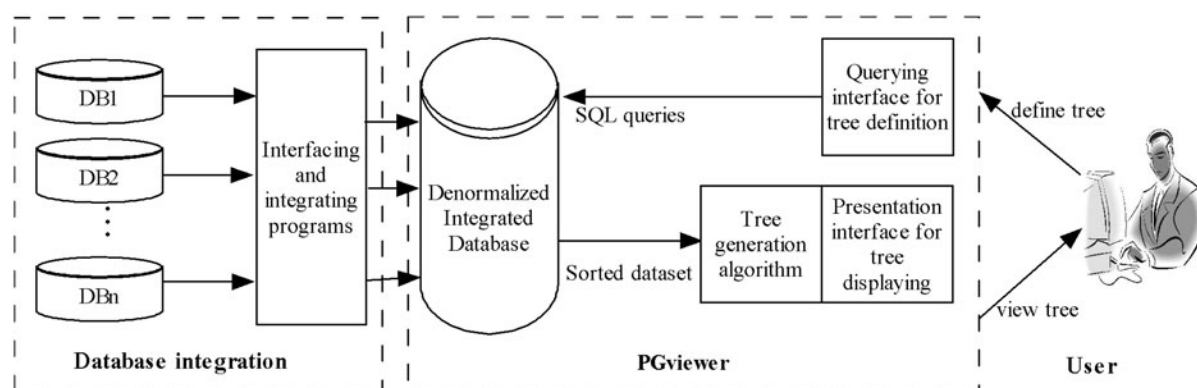


Fig. 1. Architecture of the Phenogenes Viewer.

```
<gene v="tuberous sclerosis 2"><genemod v="heterozygote"></genemod><sid idref="s1.1.1"></sid></gene>
<phenotype v="cystadenoma"><anatomy v="kidney"><region v="bilateral"></region></anatomy><measure v="100 %"></measure><sid idref="s1.1.1"></sid></phenotype>
<phenotype v="hemangioma"><anatomy v="liver"></anatomy><measure v="50 %"></measure><sid idref="s1.1.1"></sid></phenotype>
<phenotype v="adenoma"><anatomy v="lung"></anatomy><measure v="32 %"></measure><sid idref="s1.1.1"></sid></phenotype>
```

Similarly, a PERL script is written for parsing the XML into a flat file so that it can be imported into our database. In the above example output, the **gene** *tuberous sclerosis 2*, three **phenotypes** (*cystadenoma*, *hemangioma* and *adenoma*), and their **anatomy** modifiers will be imported into the mouse genomics dataset.

- (c) UMLS–GO database. To demonstrate the possibility that our method could be used to find GO annotation terms using our NLP system's output, we incorporated a terminology mapping database developed previously, which maps 17 256 UMLS terms to GO terms (Sarkar et al., 2003). For example, the UMLS term 'apoptosis' (C0162638) is mapped on to two GO terms: 'apoptosis' (GO:0006915) and 'cytolysis' (GO:0019835) in the UMLS–GO database.

In summary, we obtained eight dimensions in our mouse genomics dataset, namely (1) PubMed_ID (2) journal (3) year (4) authors (5) gene name (6) anatomy (7) phenotype and (8) GO terms.

PGviewer

Two user interfaces, a querying interface and a presentation interface, were developed using JAVA to interact with the database. These interfaces are general in that they interact with any database that has been created. The querying interface shows users the dimensions of the databases and allows users to specify the desired dimensions and the desired clustering order. In addition, the interface generates the appropriate database queries and sends them to the database. The presentation interface processes the returned dataset using a tree generation algorithm and displays the generated tree reflecting relationships among dimensions according to user's specifications.

Querying interface A screen-shot of the querying interface applied to the mouse genomics dataset is shown in Figure 2. Users utilize this querying interface to select, arrange and apply filtering criteria to their specific dimensions that will be shown in the presentation interface later. The candidate dimensions are listed in the leftmost column automatically based on the columns of the denormalized database table. Users can select each desired dimension by pressing one of the four select buttons to the right of the column. Selected dimensions will be removed from the leftmost column and will be added in the second column, where users can arrange their order arbitrarily using the 'Up' and 'Down' buttons. This list determines the clustering order of the levels where each dimension will appear in the tree view. In Figure 2, the user chose the dimension GO term, Phenotype, Gene and PubMed ID and requested that they can be clustered in this order. In this interface, users can also specify the sorting order (i.e. ascending or descending) and apply filters for every dimension (i.e. '=', '>', '<' and 'like'). In the example in Figure 2, the GO term dimension will be filtered by 'like cell' to display GO terms containing the string 'cell'. The user could easily obtain a different view, e.g. a Phenotype–GO_term–gene–PubMed_ID view, by re-arranging the order of selected dimensions in the second column.

Tree generation algorithm The generation of a tree view makes use of the sorting and grouping function of a Database management system (DBMS) SQL query. After users choose the dimensions and filtering criteria, a SQL query representing this query is constructed automatically to fetch the data from the database. If the denormalized table in the database is called 'crosstable', the SQL query corresponding to Figure 2 will automatically generate the following query: 'select GO_term, Phenotype, Gene, PubMed_ID from crosstable where GO_term like "%cell%" group by GO_term, Phenotype, Gene, PubMed_ID order by GO_term asc, Phenotype asc, Gene asc, PubMed_ID asc'. The DBMS will sort the dataset and apply filters according to the user's definition. For example, if the original dataset in the database is as Table 1, then the retrieved dataset by the SQL query will be as in Table 2.

After retrieving the sorted dataset, an algorithm is used to merge the adjacent duplicated values of a particular dimension if the values of the previous dimension are also duplicated. In the case of Table 2, the merged data are displayed in Table 3. The corresponding tree view generated from data in Table 3 is shown in Figure 3.

Presentation interface After the generation of a tree, the presentation interface presents the tree and related information using the JTree class of the JAVA Swing package. Figure 4 shows a screen-shot of the presentation interface based on the query illustrated in Figure 2.

To speed up the response time of the presentation interface, we used a database paging technique, in which the database returns 1000 rows of data at a time. The nodes of the tree view represent the selected dimensions, including

1. Select dimensions to be shown:

Available dimensions:	Select and Order:
OMIM_ID	GO_term
Chromosome	Gene
Location	Disorder
Year	PubMed_ID
Journal	
Authors	

=> <=> =>> <<=<

Up Down

2. Define dimensions:

Sorting:

Dimension	Sorting
GO_term	ASC
Gene	ASC
Disorder	ASC
PubMed_ID	ASC

Add filter ↓

Filtering (combined by "AND"):

Dimension	Operator	Value
GO_term	like	cell

Delete

Fig. 2. Query interface showing dimensions of the Human Genomics Dataset.

Table 1. Example of an original dataset

PubMed ID	Disorder	Gene	GO term
10021369	Gardner syndrome	APC	Cell adhesion
10319819	Alzheimer disease	APP	Cell adhesion
10319819	Schizophrenia, chronic	APP	Cell adhesion
10051640	Gardner syndrome	APC	Cell adhesion
10097173	Alzheimer disease	APP	Cell adhesion
10097173	Schizophrenia, chronic	APP	Cell adhesion
1372485	Adenoma, periampullary	APC	Cell adhesion
13752393	Adenoma, periampullary	APC	Cell adhesion
...	...	...	...

Table 2. Sorted dataset by SQL query

GO term	Gene	Disorder	PubMed ID
Cell adhesion	APC	Adenoma, periampullary	1372485
Cell adhesion	APC	Adenoma, periampullary	13752393
Cell adhesion	APC	Gardner syndrome	10021369
Cell adhesion	APC	Gardner syndrome	10051640
Cell adhesion	APP	Alzheimer disease	10097173
Cell adhesion	APP	Alzheimer disease	10319819
Cell adhesion	APP	Schizophrenia, chronic	10097173
Cell adhesion	APP	Schizophrenia, chronic	10319819
...	...	...	...

their values and the numbers of their direct children nodes (within the brackets). Users can expand and collapse a node by clicking the '+' or '-' signs in the beginning of the node. This presentation interface also provides a detailed view of some special dimensions. For example, if a node associated with a

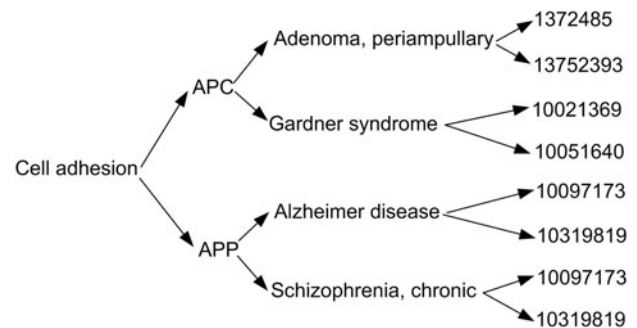


Fig. 3. Tree generated from data in Table 3.

Table 3. Merged dataset of Table 2 showing production of a tree

GO term	Gene	Disorder	PubMed ID
Cell adhesion	APC	Adenoma, periampullary	1372485
			13752393
		Gardner syndrome	10021369
			10051640
	APP	Alzheimer disease	10097173
			10319819
		Schizophrenia, chronic	10097173
			10319819
...	...	...	...

PubMed ID is selected, the corresponding article's title and abstract will be displayed in a text area or in a popup window (The popup window is not shown in Fig. 4). For the mouse genomic dataset, the relationships between gene and phenotype information extracted by BioMedLEE are shown in the table below

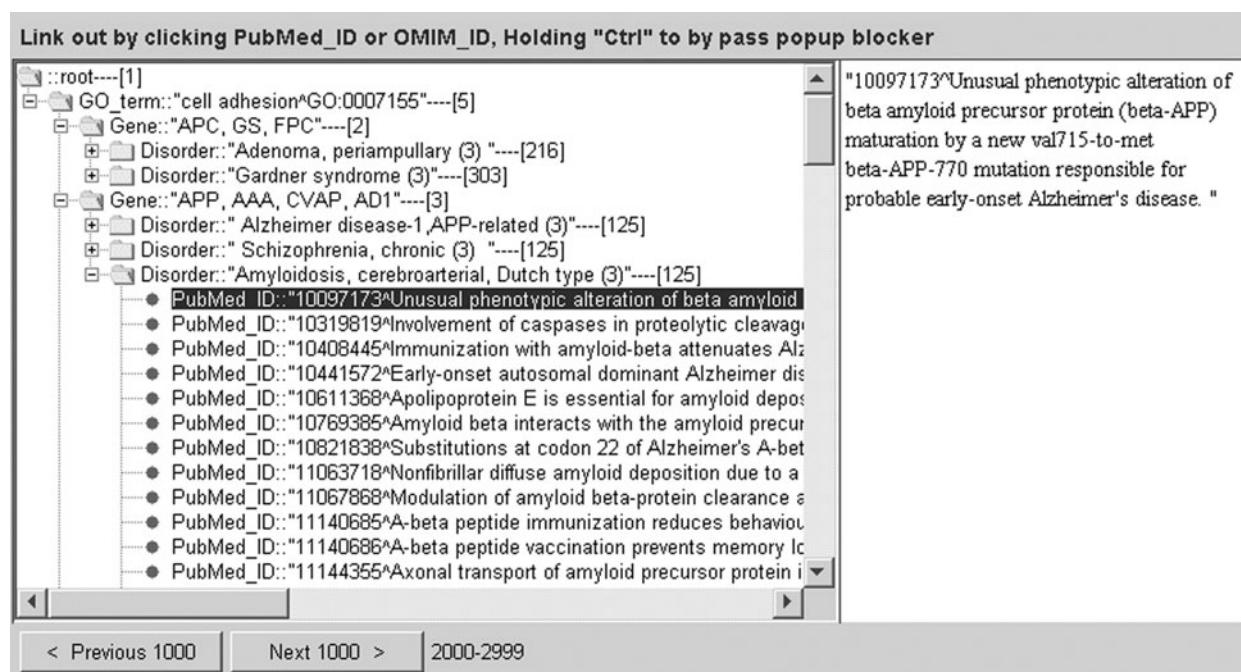


Fig. 4. Query results displayed in the presentation interface of Phenogenes Viewer (Human Genomics Dataset, comprehensive dataset from of Table 1, order of dimensions: GO, Gene, Disorder, PubMedID).

the text area. The table contains two columns. The first column shows paired gene–phenotype relationships based on text and the second column shows corresponding paired terminology codes. When a row representing a relationship between the gene and phenotype is selected, the corresponding original words from the titles and abstracts will become highlighted using different colors (red for gene and blue for phenotype). Thus, users can easily map the captured phenotypic information back to the original text and read the context.

The process of users' querying a tree and viewing the constructed tree can be repeated. After a user obtains a tree view in his/her first attempt, he/she wants to explore whether another view will be more helpful. Then, the user can alter the tree definition and see the modified tree view immediately or compare it with the previous tree view.

The viewer's ability to adapt to an increase in the number of dimensions is completely automatic. No modifications are needed for the query interface to display the altered schema. Whenever the centralized table is modified to contain more dimensions (table columns), the interface will automatically read all the columns and populate the list of available dimensions in the interface. A list of dimensions (metadata) is read by the viewer in order to display their names automatically in the user interface with no interventions of the user.

Evaluation methods

In order to evaluate our method, we inspected our system based on the five requirements for visualizing multidimensional genotypic and phenotypic information we presented in the Introduction section.

- (1) We investigated the ability of our tool to handle many dimensions and a large table by selecting different numbers of dimensions with and without applying filters.
- (2) We evaluated the flexibility of the querying interface by arbitrarily selecting and ordering different combinations of available dimensions and applying different filters.
- (3) We tested the flexibility of our presentation interface by comparing the human genomics dataset with its NCBI OMIM counterparts 'Search

Gene Map' and 'Search Morbid Map' interfaces in three important dimensions, *gene symbol*, *location* and *disorder*.

- (4) The capability of coordinating and integrating different databases was examined by observing if dimensions from different databases could be queried and displayed at the same time. For example, we selected *authors* from MEDLINE database, *gene* and *phenotype* from BioMedLEE and *GO_term* from GO database.
- (5) The efficiency of the user interface was estimated by measuring the approximate time for constructing a typical query.

RESULTS AND DISCUSSION

The integrated denormalized table associated with the human genomics dataset contains 739 985 rows of entries and the table in the mouse genomics dataset contains 22 271 rows. Since the query definition interface allows selecting any dimension in any order, it allows for 623 529 and 109 600 distinct dimension permutations for the human genomic dataset and the mouse genomics dataset, respectively. We utilized the DBMS of MySQL to sort the datasets. The database schema is straightforward and consists of a single denormalized table. Therefore, the scalability of the database mainly depends on the DBMS' capability of sorting and querying a single large table. Having only one table improves performance by eliminating the need to join different tables, and by simplifying integration of the DBMS with the viewer component. This strategy does not limit the flexibility or the scalability: new dimensions are easily accommodated by adding new fields to the denormalized table. We acknowledge the drawback of the maintenance of the system, as new dimensions would require recompiling the complete denormalized database; additionally as many more dimensions are added it is likely that the database queries would be less efficient. With no changes in

dimensions, updating the current databases may require as much as 15 man-hours.

- (1) When the maximum number of dimensions was used without any filters, PGviewer returns and displays the results in ~5 s in the mouse genomics dataset and 80 s in the human genomics dataset. In other conditions, the response time of the system may vary according to the number of rows of the table, the number of selected dimensions, and specific filters and may reach as much as 160 s. Generally, large row numbers and dimension numbers will slow down the system. On the other hand, the use of filters will speed up the response.
- (2) The query interface successfully executed all the queries formed by arbitrarily selecting dimensions, ordering them and applying various filters. In both datasets, the presentation interface displayed the corresponding structured tree views correctly.
- (3) For three important dimensions in OMIM (*gene symbol*, *location* and *disorder*), 'Search Gene Map' provides a pre-defined table in its output that aligns the three dimensions in the order of *location*, *gene symbol* and *disorder* and is sorted by *location* alphabetically. 'Search Morbid Map' provides a pre-defined table in the order of *disorder*, *gene symbol* and *location* and is sorted by *disorder*. However, there could be other dimension orders that are of importance to biologists. For example, the order of *location*, *disorder* and *gene symbol* will cluster *disorders* under specific *locations*. Thus, hotspot of certain diseases can be discovered easily. This cannot be obtained from OMIM interfaces directly. In contrast, PGviewer could visualize this ordering gracefully in expandable trees (Fig. 5). Figure 5 illustrates the clustering of the disorder 'breast cancer' under specific locations found in OMIM and presented in PGviewer. It is clear that chromosome 17 is a hotspot location for breast cancer. In addition, PGviewer can help in discovering unnoticed new knowledge buried across multiple databases. In this paper, the inclusion of GO provides possible molecular mechanisms for disorders. Figure 6 shows that 'ATP binding' might be an important molecular mechanism for breast cancer.
- (4) PGviewer could successfully retrieve data from its component databases and properly visualize the result in a tree view.
- (5) For the efficiency of the user interface, we observed that typical queries took ~1 min or less to perform.

These results show that our method meets the five requirements for a flexible and generalizable information visualization tool for phenomic data as described in the Introduction section. Therefore, it could be a standard interface model for designing any model organism database, such as MGI and Flybase, because these databases actually contain similar types of information. Users need not spend additional time to learn different user interfaces in different databases. Furthermore, our method provides advantages that are absent in existing databases and could be a possible solution for database unification in the interface level in the post-genomic era.

The advantages of our method reside in two major aspects. First, it allows users' arbitrary selection and ordering of desired dimensions visually in its query interface design. This maximizes the

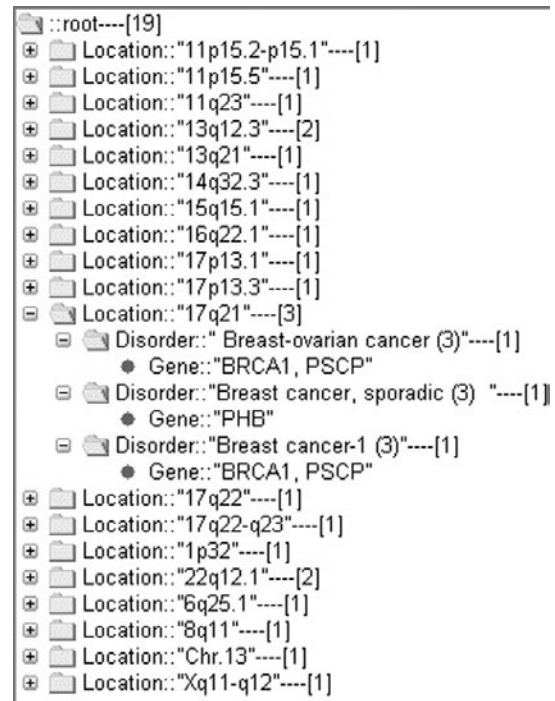


Fig. 5. Tree view of gene location, disorder and gene (Human Genomics Dataset).

flexibility of users' queries and provides improved efficiency for constructing an intended query. The deceptively simple user-interface of PGviewer conceals a powerful capability for requesting and presenting any selected permutation of dimensions. For example, the view of OMIM disorders organized according to the GO illustrated in Figure 6 is a useful presentation of the phenome, which is analogous to those presentations available in the MGI and Flybase. Since, to our knowledge, there are no browsers, which currently provide a view of OMIM disorders using a GO query, the human genomic database viewed by PGviewer proposes an original and useful functional genomic approach to organizing human phenotypes. Second, it visualizes the relationships among the informational dimensions using a hierarchical expandable tree based on user-defined queries. In a tree view, duplicate information is reduced to one node and similar information is arranged close to each other. Thus, patterns and structures of genotypic and phenotypic information can be easily perceived. In contrast, in a tabular list containing gene and phenotype relationships, the relationships would not be obvious if the table contains many entries and is not ordered. Our tool will order the list by gene and phenotype and construct a tree. Thus, associative relationships between genes and phenotypes are clear. Other advantages include the ability of handling multiple dimensions from different databases. Our method is general and can be used for any type of multidimensional data, although in this paper we focused on genotype-phenotype relationships. However, it should be noted that our visualization method assumes data integration into one database has occurred and is not aimed at a general solution for integrating heterogeneous biology databases in the level of the data source.

We realize that there are also limitations in our method. First, the new relationships found in our viewer are suggestive but not

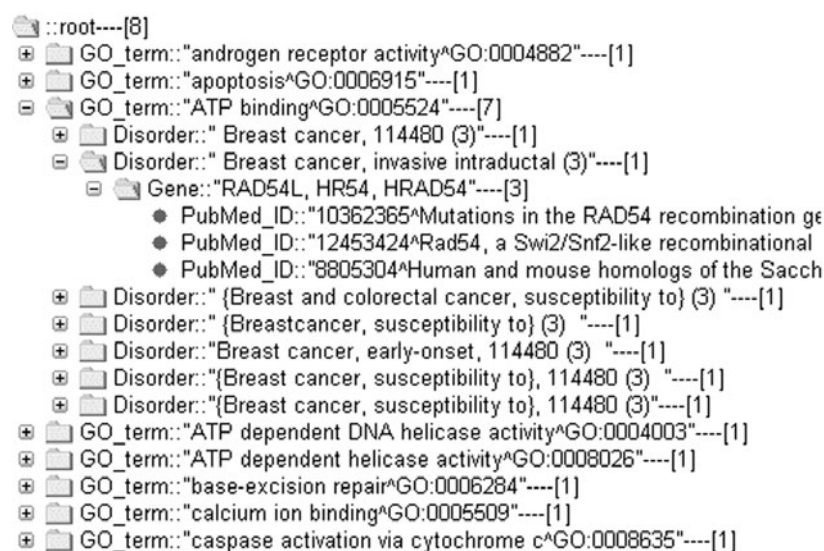


Fig. 6. Tree view of GO term, disorder, gene and PubMed ID (Human Genomics Dataset). The tree structure we use represents an ordering or clustering of information, and should not be associated with a hierarchical classification, which is a typical use of a tree when specifying an ontology or taxonomy.

confirmative because transitions of relationships may not be always correct. For example, relationships between GO terms and disorders in Figure 6 are suggestive. Locuslink provides relationships from genes to GO terms and OMIM specifies the genes associated with disorders. It is possible that only part of the GO terms defines the real molecular mechanisms for breast cancer and others are just possible mechanisms. Second, the tree view design cannot show an overview of the whole tree in one screen due to size limitations. Visualization using graphs with small size nodes, such as in some molecular networks (Koike and Rzhetsky, 2000), have been shown to solve this issue. Third, a tree view is not good for showing all the information related to a single object (node) as a graph can, because a node in a tree can only have one parent while a node in a graph can have many different parents as well as different types of relationships other than parent-child.

Our future work will involve further refinement and development of PGviewer. The many research issues we will work on will involve (1) developing a more generalizable structure for facilitating the integration of diverse databases and dimensions and (2) advancing graphical representation of the data so that many different kinds of graphs and views can be obtained.

CONCLUSION

In this paper, we have presented a novel flexible visualization tool, called Phenogenes Viewer, in response to the five basic requirements for displaying multidimensional genotypic and phenotypic information. Our work is novel in several ways. First, it allows users to dynamically specify the clustering order of data presentation so that they can focus on a view of the data that is relevant for their research interests. Second, it shows the ability to visualize structured data across different databases and ontologies including coded gene-phenotype relationships extracted from text data. Third, it provides a scalable and generalizable interface across both structured and textual databases and could be used as a standard unified interface

model for designing any model organism databases, such as MGI and Flybase. Additionally, the proposed viewer provides a seamless user interface experience across heterogeneous genomic and post-genomic databases. We believe that this method which integrates data from multiple sources and allows users to dynamically visualize the multiple dimensions, is a powerful and promising tool that should substantially facilitate biological research.

ACKNOWLEDGEMENTS

The authors thank Judith A. Blake, Janan T. Eppig and Joanna Amberger for providing assistance in understanding the MGI and OMIM genomics databases. We also acknowledge the contribution of tools or datasets provided by Jianrong Li, Hua Xu and Lyudmila Shagina. This study is partially supported by the National Institute for Allergy and Infectious Disease Grant no. 1U54 AI 57159-01 and by the National Library of medicine Grants nos R01 LM007659-01, 1K22 LM008308-01 and by the NYSTAR grant no. 5-67674.

REFERENCES

- Al-Shahrour,F., Diaz-Uriarte,R. and Dopazo,J. (2004) FatiGO: a web tool for finding significant associations of Gene Ontology terms with groups of genes. *Bioinformatics*, **20**, 578–580.
- Ashburner,M., Ball,C.A., Blake,J.A., Botstein,D., Butler,H., Cherry,J.M., Davis,A.P., Dolinski,K., Dwight,S.S., Eppig,J.T. *et al.* (2000) Gene Ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet.*, **25**, 25–29.
- Bairoch,A. and Apweiler,R. (1996) The SWISS-PROT protein sequence data bank and its new supplement TREMBL. *Nucleic Acids Res.*, **24**, 21–25.
- Baker,P.G., Brass,A., Bechhofer,S., Goble,C., Paton,N. and Stevens,R. (1998) TAMBIS—Transparent Access to Multiple Bioinformatics Information Sources. In *Proceedings of Sixth International Conference on Intelligent Systems for Molecular Biology*, Montréal, Québec, Canada, pp. 25–34.
- Benson,D.A., Karsch-Mizrachi,L., Lipman,D.J., Ostell,J., Rapp,B.A. and Wheeler,D.L. (2000) GenBank. *Nucleic Acids Res.*, **28**, 15–18.
- Bodenreider,O. and Mitchell,J.A. (2003) Graphical visualization and navigation of genetic disease information. In *Proceedings of the AMIA Symposium*, Washington DC, 792 p.

- Bult,C.J., Blake,J.A., Richardson,J.E., Kadin,J.A., Eppig,J.T., Baldarelli,R.M., Barsanti,K., Baya,M., Beal,J.S., Boddy,W.J. *et al.* (2004) The Mouse Genome Database (MGD): integrating biology with the genome. *Nucleic Acids Res.*, **32** (Database issue), D476–D481.
- Cantor,M.N. and Lussier,Y.A. (2003) Putting data integration into practice: using biomedical terminologies to add structure to existing data sources. In *Proceedings of the AMIA Symposium*, Washington, DC, pp. 125–129.
- Cantor,M.N. and Lussier,Y.A. (2004) Mining OMIM for Insight in Complex Diseases. *Medinfo 2004*, 735–737.
- Chen,I.M., Kosky,A.S., Markowitz,V.M., Szeto,E. and Topaloglou,T. (1998) Advanced query mechanisms for biological databases. *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, **6**, 43–51.
- Chen,L. and Friedman,C. (2004) Extracting Phenotypic Information from the Literature via Natural Language Processing. *Medinfo 2004*, 758–762.
- Eckman,B.A., Kosky,A.S. and Laroco,L.A., Jr (2001) Extending traditional query-based integration approaches for functional characterization of post-genomic data. *Bioinformatics*, **17**, 587–601.
- FlyBase Consortium (2003) The FlyBase database of the Drosophila genome projects and community literature. *Nucleic Acids Res.*, **31**, 172–175.
- Freimer,N. and Sabatti,C. (2003) The human phenotype project. *Nat. Genet.*, **34**, 15–21.
- Friedman,C., Alderson,P.O., Austin,J.H., Cimino,J.J. and Johnson,S.B. (1994) A general natural-language text processor for clinical radiology. *J. Am. Med. Inform. Assoc.*, **1**, 161–174.
- Friedman,C., Kra,P., Yu,H., Krauthammer,M. and Rzhetsky,A. (2001) GENIES: a natural-language processing system for the extraction of molecular pathways from journal articles. *Bioinformatics*, **17**(Suppl. 1), S74–S82.
- Friedman,C., Liu,H. and Shagina,L. (2003) A vocabulary development and visualization tool based on natural language processing and the mining of textual patient reports. *J. Biomed. Inform.*, **36**, 189–201.
- Graefe, G. *et al.* (1998) Electronic database operations for perspective transformations on relational tables using pivot and unpivot columns. Microsoft Corporation, Patent number 6298342.
- Gray,J., Bosworth,A., Layman,A. and Pirahesh,H. (1996) Data cube: a relational aggregation operator generalizing GROUP-BY, CROSS-TAB, and SUB-TOTALS. In *Proceedings of the Twelfth International Conference on Data Engineering*, New Orleans, LA, pp. 152–159, IEEE Computer Society Press.
- Hamosh,A., Scott,A.F., Amberger,J., Bocchini,C., Valle,D. and McKusick,V.A. (2002) Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders. *Nucleic Acids Res.*, **30**, 52–55.
- Hristovski,D., Peterlin,B., Mitchell,J.A. and Humphrey,S.M. (2003). Improving literature based discovery support by genetic knowledge integration. *Stud. Health Technol. Inform.*, **95**, 68–73.
- Jenssen,T.K., Laegreid,A., Komorowski,J. and Hovig,E. (2001) A literature network of human genes for high-throughput analysis of gene expression. *Nat. Genet.*, **28**, 21–28.
- Karp,P.D. (2001) Pathway databases: a case study in computational symbolic theories. *Science*, **293**, 2040–2044.
- Kasprzyk,A., Keefe,D., Smedley,D., London,D., Spooner,W., Melsopp,C., Hammond,M., Rocca-Serra,P., Cox,T. and Birney,E. (2004) EnsMart: a generic system for fast and flexible access to biological data. *Genome Res.*, **14**, 160–169.
- Koike,T. and Rzhetsky,A. (2000) A graphic editor for analyzing signal-transduction pathways. *Gene*, **259**, 235–244.
- Kolpakov,F., Ananko,E., Kolesov,G.B. and Kolchanov,N.A. (1998) GeneNet: a gene network database and its automated visualization. *Bioinformatics*, **14**, 529–537.
- Krauthammer,M., Kra,P., Iossifov,I., Gomez,S.M., Hripsak,G., Hatzivassiloglou,V., Friedman,C. and Rzhetsky,A. (2002) Of truth and pathways: chasing bits of information through myriads of articles. *Bioinformatics*, **18** (Suppl. 1), S249–S257.
- Lindberg,C. (1990) The Unified Medical Language System (UMLS) of the National Library of Medicine. *J. Am. Med. Rec. Assoc.*, **61**, 40–42.
- Liu,H. and Friedman,C. (2000) A method for vocabulary development and visualization based on medical language processing and XML. In *Proceedings of the AMIA Symposium*, Los Angeles, CA, pp. 502–506.
- Lussier,Y.A. and Li,J. (2004) Terminological mapping for high throughput comparative biology of phenotypes. *Pac. Symp. Biocomput.*, 202–213.
- Maglott,D.R., Katz,K.S., Sicotte,H. and Pruitt,K.D. (2000) NCBI's LocusLink and RefSeq. *Nucleic Acids Res.*, **28**, 126–128.
- Mahner,M. and Kary,M. (1997) What exactly are genomes, genotypes and phenotypes? And what about phenomes? *J. Theoret. Biol.*, **186**, 55–63.
- Marchler-Bauer,A., Address,K.J., Chappey,C., Geer,L., Madej,T., Matsuo,Y., Wang,Y. and Bryant,S.H. (1999) MMDB: Entrez's 3D structure database. *Nucleic Acids Res.*, **27**, 240–243.
- Rebhan,M., Chalifa-Caspi,V., Prilusky,J. and Lancet,D. (1998) GeneCards: a novel functional genomics compendium with automated data mining and query reformulation support. *Bioinformatics*, **14**, 656–664.
- Rindflesch,T.C., Libbus,B., Hristovski,D., Aronson,A.R. and Kilicoglu,H. (2003) Semantic relations asserting the etiology of genetic diseases. In *Proceedings of the AMIA Symposium*, Washington, DC, pp. 554–558.
- Rzhetsky,A., Iossifov,I., Koike,T., Krauthammer,M., Kra,P., Morris,M., Yu,H., Duboué,P.A., Weng,W. and Wilbur,W.J. (2004) GeneWays: a system for extracting, analyzing, visualizing, and integrating molecular pathway data. *J. Biomed. Inform.*, **37**, 43–53.
- Rzhetsky,A., Koike,T., Kalachikov,S., Gomez,S.M., Krauthammer,M., Kaplan,S.H., Kra,P., Russo,J.J. and Friedman,C. (2000) A knowledge model for analysis and simulation of regulatory networks. *Bioinformatics*, **16**, 1120–1128.
- Sarkar,I.N., Cantor,M.N., Gelman,R., Hartel,F. and Lussier,Y.A. (2003) Linking biomedical language information and knowledge resources: GO and UMLS. *Pac. Symp. Biocomput.*, 439–450.
- Tao,Y., Liu,Y., Friedman,C. and Lussier,Y.A. (2004) The use of information visualization techniques in bioinformatics during the postgenomic era. *Drug Discov. Today: BIOSILICO*, **2**, 237–245.
- Wheeler,D.L., Chappey,C., Lash,A.E., Leipe,D.D., Madden,T.L., Schuler,G.D., Tatusova,T.A. and Rapp,B.A. (2000) Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res.*, **28**, 10–14.
- Wong,L. (2000) The functional guts of the Kleisli query system. *ACM SIGPLAN Notices*, **35**, 1–10.
- Zeeberg,B.R., Feng,W., Wang,G., Wang,M.D., Fojo,A.T., Sunshine,M., Narasimhan,S., Kane,D.W., Reinhold,W.C., Lababidi,S. *et al.* (2003) GoMiner: a resource for biological interpretation of genomic and proteomic data. *Genome Biol.*, **4**, R28.
- Zhang,B., Schmoyer,D., Kirov,S. and Snoddy,J. (2004) GOTree Machine (GOTM): a web-based platform for interpreting sets of interesting genes using Gene Ontology hierarchies. *BMC Bioinformatics*, **5**, 16.
- Zhong,S., Li,C. and Wong,W.H. (2003) ChipInfo: software for extracting gene annotation and gene ontology information for microarray analysis. *Nucleic Acids Res.*, **31**, 3483–3486.