GENEWEBEX: GENE ANNOTATION WEB EXTRACTION, AGGREGATION, AND UPDATING FROM WEB-INTERFACED BIOMOLECULAR DATABANKS

MARCO MASSEROLI*,† ANDREA STELLA†, MYRIAM ALCALAY‡,§ and FRANCESCO PINCIROLI†

†Bioengineering Department, Politecnico di Milano, Piazza Leonardo da Vinci, 32, I-20133 Milano, Italy
‡IEO — European Institute of Oncology, I-20141 Milano, Italy
§IFOM — FIRC Institute of Molecular Oncology, I-20139 Milano, Italy
∗masseroli@biomed.polimi.it
http://www.biomed.polimi.it/BioIntro/english/people/research/Masseroli.htm

Numerous genomic annotations are currently stored in different Web-accessible databanks that scientists need to mine with user-defined queries and in a batch mode to orderly integrate the diverse extracted data in suitable user-customizable working environments. Unfortunately, to date, most accessible databanks can be interrogated only for a single gene or protein at a time and generally the data retrieved are available in HTML page format only. We developed GeneWebEx to effectively mine data of interest in different HTML pages of Web-interfaced databanks, and organize extracted data for further analyses. GeneWebEx utilizes user-defined templates to identify data to extract, and aggregates and structures them in a database designed to allocate the various extractions from distinct biomolecular databanks. Moreover, a template-based module enables automatic updating of extracted data. Validations performed on GeneWebEx allowed us to efficiently gather relevant annotations from various sources, and comprehensively query them to highlight significant biological characteristics.

Keywords: Web wrapper; data extraction; biomolecular database; genomic information; microarray data interpretation.

1. Introduction

Collaborative efforts for decoding and studying the structure of many genomes are producing a growing amount of publicly available sequence data. Numerous information describing individual genes and their encoded protein products continues to accumulate in many different and widely distributed databanks [1], where data are usually stored in sets of text files or in relational databases. Most of these biomolecular databanks are easily accessible through heterogeneous Web interfaces but require increasing expertise to be comprehensively queried. Some of them also provide a FTP access to retrieve the whole set of most of their data in a structured

*To whom correspondence should be addressed.
format, mainly inside ASCII flat files. Very few biomolecular databanks using a relational database provide a direct remote access to the used backend database due to security reasons.

New high-throughput technologies in biology — such as DNA microarrays, oligonucleotide arrays, and serial analysis of gene expression — are generating massive data sets describing the biological function of thousands of genes at once in terms of expression profiles. At present, the challenge is to find ways to exploit this huge amount of information for understanding cellular mechanisms underlying complex phenotypes. To this aim, it is necessary to provide every researcher with bioinformatics tools able to mine the increasing amount of biological information publicly available in biomolecular databanks, and automatically connect gene expression data with the mined information.

Many endeavors are being made in creating integrated access to the numerous available databanks. Several approaches have been proposed and utilized in different systems. The most popular are data warehousing (e.g. SRS [2], NCBI/Entrez [3]), multi databases (e.g. BACIIIS [4], TAMBIS [5], BioKleisli [6]), federated databases (e.g. ISYS [7], DiscoveryLink [8]), mediator based systems (e.g. BioDataServer [9]), and information linkage (e.g. GeneLynx [10], GeneCards [11], SOURCE [12], Harvester [13]). Nevertheless, most of these require resources and expertise that only companies and big research centers can afford in order to be implemented and maintained.

Other systems, such as GeneLynx, GeneCards, SOURCE, and Harvester, have been set and compiled as integrational databanks with a Web interface for providing easy query and providing simple access to multiple biomolecular information resources. Nevertheless, these systems have been created to collect and present data organized for individual nucleotide or amino acid sequences, and some of them (e.g. GeneLynx) only present a series of links to other external resources. Moreover, they are designed for human browsing and not for machine reading. Thus, these and other valuable databanks poorly adapt to biological interpretative analyses and knowledge discovery from vast data sets, which involve comparative evaluation of multiple features of many nucleotide and/or amino acid sequences at once. Hence, large data set interpretation requires the support of new automated tools for mining and comparing the information of interest from the databanks where it is available.

Accessing the data in the structured form they have inside the databank would be the best option to mine a biomolecular databank. Nevertheless, to this aim the only FTP access that some biomolecular databanks provide to their structured data obliges to locally reimplement and maintain the whole databank. This requires many expertise and resources that only big research centers and companies can afford. Moreover, usually the annotations required for the proper biological analysis of high-throughput experiment results reside in several different biomolecular databanks, some of which do not provide access to their structured data. Thus, even locally reimplementing and maintaining many different databanks could not be enough for obtaining all available data necessary for the appropriate analysis of
high-throughput experiment results. On the other hand, generally researchers do not need all the data present in a databank but only a subset, either regarding number of genes or annotations of each gene. The best technological option would be accessing the data in XML format through Web services but at present still very few data providers offer access to their data via a Web services model.

Here we report an automatic method for extracting selected data of multiple nucleotide and/or amino acid sequences from different biomolecular databanks accessible through Web interfaces, organizing extracted data in order to allow their integration to expression profiling results, and performing comparisons and further analyses on them. The method has been implemented in a prototype software package called GeneWebEx, i.e. Gene annotation Web Extractor (http://www.bioinformatics.polimi.it/GeneWebEx/).

General applicability, simple use, versatility, and effectiveness make this software potentially useful for individual researchers and small laboratories that do not have the informatics expertise and funding for developing personal applications or maintaining the powerful but expensive commercial packages.

2. Methods

In GeneWebEx, by using Java programming language, we implemented the mining method which we developed for automatically extracting, structuring, and storing in aggregated form data of interest from HTML pages of biomolecular databanks. The method works as described in Fig. 1. HTML pages containing data of interest are retrieved from biomolecular databanks accessible via Web. Using the Document Object Model (DOM) recommendation of the World Wide Web Consortium (http://www.w3.org/DOM/), each retrieved HTML page is parsed to separately identify data, HTML tags, and other page elements such as Javascript functions or comments. Tags and data are considered to represent the HTML page with a hierarchical tree structure, i.e. a data structure composed of nodes containing either a

![Fig. 1. Steps of the implemented method for automatically extracting and aggregating data from different HTML pages of diverse biomolecular databanks.](image)

Fig. 2. Example of a HTML page (A), its HTML code (B), and the corresponding data tree structure (C).

HTML tag (tag node), or data inside the page (data node) (Fig. 2). Using templates previously created on HTML pages with similar structure, the developed mining method automatically locates and extracts the different data of interest inside the created hierarchical tree structure of each retrieved HTML page. Data extracted from multiple pages, also of different biomolecular databanks, are structured and stored aggregated.

Each used template provides the mining method with the information for:

1. accessing the HTML pages of a biomolecular databank containing data of interest;
2. locating in a HTML page the exact position of data of interest, identified by specific HTML tags;
3. extracting these data and those identified by the same HTML tags present in other pages with similar structure;
4. performing filtering operations on the extracted data, if needed, and using some data to characterize the others.

2.1. Creation of templates for HTML page data extraction

Templates are created according to a specific developed algorithm and user interaction. For each template the user must provide some sequences of characters, selected on a reference HTML page, which can identify the data to extract. Two template creation modalities have been defined: semiautomatic and manual.
Fig. 3. Example of GeneWebEx extraction from the UniGene databank. A reference UniGene HTML page is where GeneWebEx allows visual selection of the data of interest. The anchor (e.g. “SIMILARITIES”) and other two sequences of characters (e.g. “H.sapiens” and “asparagine”) are selected to identify the HTML tags containing the data to extract. In the example, the data of interest to extract are inside the dashed line. The selected sequences are highlighted inside solid line boxes.

2.1.1. Semiautomatic creation modality

This procedure enables creating templates for extracting groups of data structured as they appear formatted on a HTML page. For each set of data to extract, the user must select three sequences of characters on a reference HTML page (Fig. 3). The first sequence must be an anchor, i.e. a unique sequence of characters on the page. It represents a landmark for locating inside a page the HTML tag containing the data of interest to extract. The other two sequences of characters must be selected among the data to extract. Within the tree structure representation of the reference page, they identify the tag node (father node) containing in its sub-tree the data to extract.

Using the three selected sequences of characters, the algorithm automatically finds the father node of the sub-tree structure containing the data to extract, and defines the relative path in the page tree structure from the anchor data node to the father node. Then, it creates the template that includes the anchor, the HTML...
Fig. 4. Example of template creation: the selected anchor sequence of characters, the HTML page sub-tree structure with the data to extract (inside the dashed line), and their relative path vector for the example in Fig. 3. Ellipses and rectangles represent tag nodes and data nodes of the page tree structure, i.e. tree nodes containing HTML tags (e.g. HTML, P, TABLE, TR, TD) and data inside the HTML page, respectively. UP, UP, +1 are the steps required to navigate the tree structure from the anchor to the sub-tree structure with the data to extract.

tag in the father node — used for correctness checking of the extracted data — and the relative path in form of path vector (Fig. 4).

Besides the data of interest, inside the sub-tree structure of the father node identified by a semiautomatically created template there are HTML tags visually formatting these data on the HTML page. According to the meaning given to each of these tags, data can be differently extracted and structured inside an extraction result table (Fig. 5A). To this aim, the developed mining method considers the information in an extraction configuration text file,
associated with every template, which users can customize according to specific mining requirements (see the User Guide section of the GeneWebEx Web site at URL http://www.bioinformatics.polimi.it/GeneWebEx/).

When not all extracted data in the sub-tree structure of the father node identified by a semiautomatically created template are of interest for the specific extraction requirements, a further template creation step allows defining filtering operations on the extraction result table. These enable considering the values in a result table column as labels, and filtering the extracted data according to selected label values only (Fig. 5). Besides, when the cells in a table row contain the name of the column they belong to, these names can be automatically assigned to the extraction result table columns, better characterizing the contained extracted data and enabling subsequent specific queries on them.

2.1.2. Manual creation modality

This procedure allows generating templates for extracting and structuring sparse data from a HTML page, independently of the way they appear formatted on the page. Templates created through the manual modality are composed of extraction units, one for each data to extract from the page, representing the columns of an extraction result table. For each extraction unit, the user must select two sequences of characters on a reference HTML page. The first sequence constitutes an anchor, i.e. a label characterizing the data to extract and representing a unique landmark in the page for locating the data father node. The second sequence of characters represents the data of interest and enables identifying the father node of the data to extract within the tree structure of the reference page. Using the two selected sequences of characters, the algorithm automatically defines the relative path in the page tree structure from the data node containing the anchor to the father node identifying the data of interest. Then, it creates the template that has the same structure of the templates created with the semiautomatic modality.

Templates created with the manual modality define names — the labels of the data to extract — and number — the extraction units — of the columns in the extraction result table. Hence, they completely identify the data to extract and how to structure them.

2.2. Extraction of data from multiple HTML pages and biomolecular databanks

The created templates for automatically extracting data of interest from an HTML page can be used to extract the same type of data from different HTML pages of the same biomolecular databank. Nevertheless, in order to obtain correct extractions, some hypotheses on the structure of the databank Web pages need to be satisfied:

— The character sequence selected as anchor must be present and identical in all HTML pages containing the data of interest.
In all databank HTML pages, the HTML tag in the father node of the sub-tree structure containing the data to extract must not change.

The defined anchor, relative path, and father node HTML tag stored inside the template to use must univocally identify the data of interest.

Between the anchor and the father node identifying the data to extract, all databank HTML pages must have an unvaried structure, i.e. the relative path must not change.

Providing to select an adequate anchor, the HTML pages of biomolecular databanks satisfy all these hypotheses. In fact, these pages are dynamically and automatically created from the data contained in the database underneath. Thus, all of them have a similar structure. Besides, the most restrictive hypothesis of invariable structure between the anchor and the father node of the data to extract is not relevant when the anchor is correctly selected inside or just beside the father node sub-tree structure containing the data of interest.

Using different templates and anchors we performed many tests on the HTML pages of the UniGene [14], LocusLink [15], Swiss-Prot [16], SOURCE [12], and GeneCards [11] databanks. Test results demonstrated that only in rare cases — with anchors selected not beside the father node sub-tree structure containing the data to extract — changes in structure between the selected anchor and the father node can occur in some pages. In these cases, the controls implemented in the extraction algorithm (e.g. correctness checking of the HTML tag in the father node) prevent from mining data not of interest. Moreover, when some data of interest are not found in a HTML page, a reference to that page and a possible reason for the unsuccessful extraction are noted in a log file. This gives to the user the chance of carefully revising only those few HTML pages possibly presenting problems during the automatic extraction.

Usually, each HTML page of a biomolecular databank contains all the information present in the databank about a single nucleotide or amino acid sequence, and can be retrieved using the identification code of that sequence. Therefore, providing the identification codes of nucleotide or amino acid sequences of interest and using the information contained inside a specifically created template, the developed method can automatically extract data of interest from multiple HTML pages of a biomolecular databank. To achieve extractions of different data from multiple HTML pages of a single or distinct biomolecular databanks, different templates can be created and automatically applied in sequence. All extracted data are aggregated and stored either in tab-delimited text files (which can be directly imported in other programs such as Microsoft Excel), or in a relational database, allowing both simple analyses and articulated queries on all aggregated data.

2.3. System validation

GeneWebEx was evaluated using a set of 729 clones resulted from the analysis of microarray experiments aimed at identifying genes that are differentially expressed.
in U937 cells after 4 hours of treatment with $10^{-6}$ M Retinoic Acid (RA). As following described, the identified putative RA target genes were classified by mining for their annotations using GeneWebEx. To verify if the extracted data were sufficient and accurate, the putative RA target genes were also independently analyzed. Literature search was performed for all known genes in the clone set considered and the obtained results were compared with those found through GeneWebEx extractions.

3. System Implementation and Results

The GeneWebEx prototype software package implements the developed method for extracting information from HTML pages of different Web-interfaced databanks and allows local aggregation and comprehensive analyses of all extracted data. The main characteristics of GeneWebEx are:

(1) a Graphic User Interface with intuitive windows for an easy use adequate to biologists and physicians (Fig. 6);
(2) a module for template creation from any reference HTML page;
(3) a module for automatic extraction of data from different HTML pages, which

![GeneWebEx Graphic User Interface](image-url)
includes an extraction delay procedure that controls the interval between subsequent requests to a Web-interfaced databank in order to avoid its overloading;

(4) parametric functioning, which adapts extraction performances by modifying parameter values inside extraction configuration text files;

(5) log files containing information about the performed data extractions and allowing quick evaluations of results;

(6) aggregation and storage of all extracted data, either in tab-delimited text files or in a relational database;

(7) a software agent module for updating the extracted data stored in the database.

3.1. Database of extracted data

We designed a relational database that can aggregate and store all the heterogeneous data extracted from different HTML pages and databanks. The database schema (Fig. 7) is composed of four tables. The table MAIN stores the general information related to each extraction (i.e. extraction date, used template, name of mined biomolecular databank, used ID code of the considered nucleotide or amino acid sequence [e.g. GenBank accession number, Clone ID, UniGene Cluster ID, LocusLink ID, Swiss-Prot accession number]). Two different tables, DATA and LINKS, contain the extracted information concerning textual data or links, respectively. Table COLUMN_NAMES stores the names of the extraction result

Fig. 7. Entity-Relation diagram of the database aggregating and storing the extracted data.
table columns, defined as described in the Methods section, which characterize the extracted information.

The designed database schema enables to comprehensively query all gathered data. In fact, each datum in the database is identified by the ID code of the nucleotide or amino acid sequence the datum refers to and that was used to extract the datum (e.g. the GenBank or Swiss-Prot accession number ID code). Besides, each extracted annotation is characterized by the name of the extraction result table column (stored in the database COLUMN_NAMES table) the annotation belongs to, and by the other data extracted with the annotation itself. When different types of ID codes for the same sequences must be used to mine data from different databanks, they can be extracted from one of the public databanks providing the ID codes of a nucleotide or amino acid sequence in different resources (e.g. GeneCards or SOURCE databanks). These different sequence ID codes are used inside the extracted data database to link annotations mined from different resources.

In the extracted data database provided with GeneWebEx, we created two sets of queries in Structured Query Language (SQL), easily accessible through a graphic interface (see Mining Results section of GeneWebEx Web site). The first set includes general purpose queries, which allow simple data mining of the annotations in the database just by specifying keywords to look for and/or type of annotations among which to search. The second set comprises articulated queries specifically designed to provide comprehensive integrated views on the annotations mined from UniGene, LocusLink, Swiss-Prot, SOURCE, and GeneCards databanks (see Mining Results section of GeneWebEx Web site).

3.2. Updating of extracted data

We used Java programming language to create a software agent module for updating the information stored in the database of the extracted data. The software agent utilizes the defined templates used to populate the database, and an identification code list of the nucleotide and/or amino acid sequences whose information need to be kept updated. At predefined intervals of time, the software agent autonomously and automatically applies the extraction rules stored inside the templates, extracts the available data of interest from the Web pages of the sequences identified in the code list, and stores them in the database, in this case replacing their old versions. Thus, also retrieved annotations presenting high temporal variability can be kept updated and synchronized to those in the original databanks.

3.3. System validation

Using a set of 729 clones identified through gene expression microarray experiments as described in the Methods section, we tested the efficacy and utility of GeneWebEx by mining specific information regarding the identified clones from different biomolecular databanks, and by analyzing the extracted annotations to functionally classify the clones.
We used GeneWebEx to mine the descriptions and identification codes in several resources, and the genomic, proteomic, cytogenetic, phylogenetic, expression, structural, functional and disease annotations for the 729 putative RA target genes. As clone identifications we used the GenBank accession numbers provided with the microarray results to mine the UniGene, SOURCE, and GeneCards databanks. Whereas, we utilized the LocusLink IDs and Swiss-Prot accession numbers extracted from the SOURCE databank to mine the LocusLink and Swiss-Prot databanks, respectively. Complete tables of extracted annotations for the identified differentially expressed genes, obtained by applying the articulated queries created in the extracted data database, are available in the Mining Results section of GeneWebEx Web site.

To evaluate correctness and efficacy of the implemented mining method, the automatically extracted annotations were visually compared with those in the HTML pages of the considered databanks. According to the defined extraction templates, we looked for false positive and false negative mined data, i.e. extracted data not of interest and data of interest present in the HTML pages but not extracted, respectively. We found that no irrelevant annotations were extracted (no false positives), and few annotations of interest present in the databank HTML pages were not extracted (few false negatives). The latter were not extracted mainly because of syntax errors or diverse incorrect structures in the HTML code of the Web page sections where they were located. Therefore, GeneWebEx proved efficient in specifically and rapidly mining the requested information for virtually all of the genes for which such information was actually available.

4. Discussion
In the last years, several efforts have been made to effectively exploit the increasing amount of information sparsely contained inside many heterogeneous biomolecular databanks accessible through Web servers [2-12]. Nevertheless, the solutions proposed to extract the information contained in different databanks and to execute comparisons, either require advanced informatics knowledge and significant resources to be implemented (e.g. legacy systems such as SRS, BioKleisli, DiscoveryLink, or BioDataServer), or only partially solve individual problems (e.g. integrational databanks such as GeneCards, SOURCE, or Harvester). The former are adequate for big research centers but not for the needs of small research laboratories or individual researchers. The latter, represented by centrally curated and publicly accessible resources, are useful for retrieving information from the only databanks they curate. Besides, they either prevent users from performing batch queries on multiple nucleotide or amino acid sequences simultaneously, or enable aggregation of some limited information only. Furthermore, in almost all cases the retrieved data are published inside HTML pages, i.e. in a format not suitable to store and structure them for further mining and analysis. On the other hand, direct access to biomolecular data in their structured form is provided by some databanks
only and only via FTP, generally to a whole set of text files containing most of the
databank data. This requires adequate expertise and resources for locally reim-
plementing and maintaining several biomolecular databanks to have a chance to
comprehensively browse and mine their data in order to get most of the annota-
tions useful for proper biological analyses of high-throughput experimental results.
Direct access to data in XML format, provided via Web services, would be the best
option. Unfortunately, at present still very few providers offer a proper data access
via a Web services model. Thus, currently the option of extracting data from data-
bank Web interfaces remains attractive, especially for researchers without extended
informatics knowledge and with limited supporting resources.

Screen scraping and wrapper software have been developed and implemented to
extract the data contained in a HTML page and to organize them in other formats
[17,18]. Nevertheless, these software solutions are based on script programs that are
generally not easily applicable to HTML pages with a complex structure, such as
those of some integrational databanks, and whose script code needs to be modified
when the page structure or simply the extraction needs change. Therefore, they are
suitable for central bioinformatics facilities rather than for customization directly
performed by the end user.

4.1. Software prototype and testing

With the above in mind, we created GeneWebEx. Its characteristics make the im-
plemented prototype a flexible and adaptable instrument that does not require pro-
gram code modifications. Users with basic informatics knowledge can easily handle
the prototype, create and configure templates according to specific data mining
needs, use them to automatically extract data from different HTML pages also of
different biomolecular databanks, and comprehensively query the retrieved data. In
case the HTML page structures of the considered biomolecular databanks change,
or mining requirements vary, the templates can be easily and quickly regenerated.
More expert users can modify the template configuration parameters, involving
knowledge of HTML language, to optimize the mining performance in relation to
the specific extraction.

As the performed tests demonstrated, the controls implemented in the mining
method prevent from inserting in the extracted data database information not of
interest and give users the chance of carefully revising only those few HTML pages
that could have presented problems during the automatic extraction, i.e. possibly
containing annotations of interest not automatically extracted.

Recently, besides some Web-interfaced integrational databanks, also a few tools
such as Ensembl EnsMart [19] have been developed for directly querying genomic
information and integrating query results with user data. Nevertheless, all these
valuable instruments give access only to the information their curators maintain.
Whereas, thanks to the use of extraction templates that each user can visually
create according to specific data extraction needs, GeneWebEx enables to extract,
aggregate, and query virtually whatever required data of interest from any Web-interfaced databank.

All these features make GeneWebEx a flexible tool especially adequate for the needs of small and medium research laboratories, which often do not have the expertise and resources to manage instruments that are more sophisticated but also more expensive and complex to use.

4.2. Extracted data database and updating software agent

The designed database schema ensures adequate flexibility for storing and aggregating data and links extracted from different HTML pages — and from distinct databanks — together with general information on the performed extractions. It also enables combining the extracted annotations with other data from diverse sources (e.g. in-house produced gene expression profiles) and executing articulated queries to further mine all the aggregated data. Thus, the designed extracted data database allows researchers to execute information comparisons and knowledge data mining analyses that are otherwise difficult to perform.

The high temporal variability of the data contained in many biomolecular databanks requires an equally high updating frequency of the data extracted and stored inside the database of the extracted data to prevent the latter from rapidly becoming obsolete. The software agent developed for updating the extracted data achieves this goal autonomously and intelligently, and provides the data aggregated and structured inside the GeneWebEx database with the fundamental characteristic of being kept up to date.

4.3. Validation and applications

We used GeneWebEx in a model system of RA-induced differentiation to demonstrate the efficiency and versatility of the proposed extraction method. It also showed the utility and potential of the implemented software to help interpreting results from differential gene expression experiments. In fact, GeneWebEx enables to enrich a list of differentially expressed clones with annotations extracted from different biomolecular databanks, freely chosen by the user according to the requirements of specific experimental objectives. This allows creating tables that enhance significant characteristics of the considered set of clones by integrating and organizing the mined annotations. Some of these tables can contain extremely valuable information for a researcher. Examples are the tables of protein similarities in different organisms, protein structures and functions (including domains, subcellular locations, and pathways), and related phenotypes and diseases (see Mining Results section of GeneWebEx Web site). At present, to our knowledge these data are not easily obtainable in such an aggregated way with any other publicly available resource.

Moreover, because GeneWebEx enables mining the same type of annotations from various sources, it allows either comparing equivalent annotations in order to
detect differences among sources, or integrating similar information to increase the number of annotated clones. This can be very useful especially for functional or disease annotations that are still scarce and not homogeneously represented in the diverse sources. In annotating our chosen data set, by interrogating different sources we could coherently classify a significant number of genes as involved in relevant functional pathways. All annotations extracted with GeneWebEx for a set of RA-regulated genes were in agreement with the results of a gene-by-gene literature search independently performed on the same gene list. However, using GeneWebEx we could extract relevant information for our chosen data set within a few minutes.

5. Conclusions

GeneWebEx represents a powerful and user-friendly tool for extracting several annotations (e.g. genomic, proteomic, cytogenetic, phylogenetic, expression, structural, functional, disease) of multiple genes from distinct biomolecular databanks accessible via Web interfaces, allowing their integration to expression profiling results. GeneWebEx characteristics make it aimed at researchers without extended informatics knowledge and with limited supporting resources, providing them with the functionalities they need to easily exploit the numerous valuable biomolecular information today sparsely available and without requiring local reimplementation of the databanks to mine. Of particular interest for the interpretation of large data sets, GeneWebEx flexible extraction of annotations allows collecting and organizing functional information of user selected gene products, which is becoming increasingly available in public databanks. This facilitates a genomic approach to the understanding of the fundamental biological processes and complex cellular patho-physiological mechanisms.

References


