International Journal of Biological, Life and Agricultural Sciences
ISSN: 2415-6612
Vol:1, No:1, 2007

Computational Method for Annotation of Protein Sequence According to Gene Ontology Terms

Razib M. Othman, Safaai Deris, and Rosli M. Illias

***Abstract*—**Annotation of a protein sequence is pivotal for the understanding of its function. Accuracy of manual annotation provided by curators is still questionable by having lesser evidence strength and yet a hard task and time consuming. A number of computational methods including tools have been developed to tackle this challenging task. However, they require high-cost hardware, are difficult to be setup by the bioscientists, or depend on time intensive and blind sequence similarity search like Basic Local Alignment Search Tool. This paper introduces a new method of assigning highly correlated Gene Ontology terms of annotated protein sequences to partially annotated or newly discovered protein sequences. This method is fully based on Gene Ontology data and annotations. Two problems had been identified to achieve this method. The first problem relates to splitting the single monolithic Gene Ontology RDF/XML file into a set of smaller files that can be easy to assess and process. Thus, these files can be enriched with protein sequences and Inferred from Electronic Annotation evidence associations. The second problem involves searching for a set of semantically similar Gene Ontology terms to a given query. The details of macro and micro problems involved and their solutions including objective of this study are described. This paper also describes the protein sequence annotation and the Gene Ontology. The methodology of this study and Gene Ontology based protein sequence annotation tool namely extended UTMGO is presented. Furthermore, its basic version which is a Gene Ontology browser that is based on semantic similarity search is also introduced.

***Keywords*—**Automatic clustering, Bioinformatics tool, Gene Ontology, Protein sequence annotation, Semantic similarity search.

I. Introduction

THE Gene Ontology (GO; http://www.geneontology.org/) is a collection of nearly 22,600 terms to describe gene and gene product attributes in any organism. The terms are structured, controlled vocabularies and organized as a Directed Acyclic Graph (DAG) in three aspects: cellular component, biological process, and molecular function. The GO is an emerging ontology that is gaining momentum for the purpose of genome, EST (expressed sequence tag), and protein annotations. The advantages of using the GO are:

1) The GO data is dynamic and constantly evolves according to the current state of biological knowledge advances.
2) The GO data is publicly available and can be downloaded at any time on the World Wide Web (WWW) in various formats that can be understandable and processable by human and machine alike.
3) The common GO terms shared by gene and protein sequences in multiple organisms in different databases can facilitate uniform queries across them.
4) The association of GO terms with nearly 2.32 million gene products that are supported by citation and evidence can affirm its reliability for future evaluation and use.

Therefore, the purpose of this study is to develop a new GO-based method to annotate protein sequences. Specifically, this study focuses on techniques to split the monolithic GO RDF/XML file and to search for semantically similar GO terms. This study also considered the development of automated tool to demonstrate the effectiveness of the proposed method.

This paper is organized as follows: the rest of Section I elaborates the problems and the proposed solutions together with the objectives, scope, and significance of this study. Section II and Section III describe the protein sequence annotation and the GO respectively. Section IV reviews the algorithms for splitting the GO RDF/XML file and searching the GO terms including the GO-based tools for annotating protein sequences. Section V explains the methodology used in this study. The summary is given in Section VI.

A. The Macro-Problem and the Proposed Solution

Application of the GO terms to annotate protein sequences is not easy, especially for species not yet inserted in public biological databases. Furthermore, for bioscientists with little computational knowledge or limited facilities it is a hard task to annotate those protein sequences. This is due to the fact that generally the existing GO-based tools are:

1) Dependent on BLAST (Basic Local Alignment Search Tool) which is computationally intensive and requires

Manuscript received January 8, 2007. This material is based upon work supported by the Malaysian Ministry of Science, Technology, and Innovation (MOSTI) in part under Intensification of Research in Priority Areas (IRPA) grants (Project No. 04-02-06-10049-EAR and 04-02-06-10050-EAR) and in part under Short Term Research (STR) grant (Project No. 75162).

Razib M. Othman is with the Faculty of Computer Science and Information Systems, Universiti Teknologi Malaysia, 81310 UTM Skudai, Johor, MALAYSIA (corresponding author; phone: 607-5532358; fax: 607-5565044; e-mail: razib@utm.my).

Safaai Deris is with the School of Graduate Studies, Universiti Teknologi Malaysia, 81310 UTM Skudai, Johor, MALAYSIA (e-mail: safaai@utm.my).

Rosli M. Illias is with the Faculty of Chemical and Natural Resources Enginneering, Universiti Teknologi Malaysia, 81310 UTM Skudai, Johor, MALAYSIA (e-mail: r-rosli@utm.my).

high-cost and high-specification hardware.
2) Dependent on RDBMS (Relational Database Management Systems) which requires the user to setup the RDBMS software and to import the data or sources into the RDBMS format.
3) Partially based on the GO data and requires the user to download the GO Annotation (GOA) data or protein sequence data sets from several sources.
4) Sequence alignment is performed to all protein sequences, but not only to sequences that indicate higher similarity.

Therefore, in this study, a new way of applying the GO terms to annotate protein sequences is introduced. The method works with three main parts. In the first part, the single GO RDF/XML file is split into smaller files. The ideas are to avoid dependency on RDBMS format, to fully use the GO data by adding the GOA data and the protein sequence data sets into the files, and to make it easier to be accessed and processed. In the second part, search is performed over the smaller GO RDF/XML files. The target is to find a group of GO terms with higher term similarity score to a GO term which is foreseen to have higher relationship with the query protein sequence. Lastly, the results obtained from the previous part are verified by computing sequence alignment score between the query protein sequence and all sequences attached to those terms. With this method, sequence alignment is carried out only to protein sequences with higher outguessed similarity. Hence, demand for high computational facilities and execution time can be reduced.

B. The Micro-Problem and the Proposed Solution

The proposed GO-based method as described in the previous subsection lead to more technical and theoretical problems. These micro-problems are related to automatic clustering and semantic similarity searching. Automatic clustering is an unsupervised learning problem that tries to divide a set of elements into a number *k* of clusters. Thus, elements in the same cluster are as similar as possible and elements in different clusters are as dissimilar as possible. Determining the number *k* of clusters is done by the algorithm and it can be regarded as a hard algorithmic problem. To cluster the GO terms into the number *k* of clusters in order to split the monolithic GO RDF/XML file, the following questions need to be resolved:

1) What is the most appropriate clustering algorithm that provides optimal solution and offers reasonable amount of processing time?
2) What is the accurate measurement for identifying the number *k* of clusters and for valuating the quality of those clusters?

A genetic split-merge algorithm that combines the parallel genetic algorithm with the split-and-merge algorithm is introduced. The algorithm works by decomposing the GO terms into a number of clusters and then automatically combines these clusters in several iterations until the best number *k* of clusters is found. The algorithm uses cohesion-and-coupling metric to measure the goodness of the generated clusters.

On the other hand, semantic similarity searching relates to the problem of determining semantic relatedness between terms either by virtue of their likeness (*bank-trust company*), synonymy (*car-automobile*), meronymy (*computer-keyboard*), antonymy (*rich-poor*), functional relationship (*marker pen-white board*), or frequent association (*orang utan-Borneo*). In the case of searching for semantically similar GO terms, they are related according to "association"; a table storing information that is shared among the GO terms. Particularly, this table provides an annotation record that is basically a link between a gene product and a GO term. To search the GO terms, the following questions need to be responded:

1) What is the most appropriate search algorithm that provides feasible solution and offers reasonable amount of execution time?
2) What is the accurate measurement for this biology-related search for measuring the semantic similarity between the GO terms?

A genetic similarity algorithm is proposed by incorporating the parallel genetic algorithm with the semantic similarity measure algorithm. The parallel genetic algorithm is used to generate solution consisting of a set of terms that best match to the user's query and to accelerate the search that involves large dimension. In the meantime, semantic similarity measure algorithm is added into the parallel genetic algorithm to measure the similitude strength between terms during the initiation of chromosome and calculation of fitness value.

C. Objective of the Study

The goal of this study is to develop a computational method to annotate protein sequences using knowledge in the GO. Therefore, this study has several objectives to achieve as follows:

1) To develop an automatic clustering algorithm using genetic split-merge algorithm in order to split the monolithic GO RDF/XML file.
2) To develop a similarity search algorithm using genetic similarity algorithm in order to find a group of semantically related GO terms.
3) To develop a tool as a proof-of-concept study that applied both algorithms mentioned above in order to highlight the capabilities of the proposed method.

D. Scope and Significant of the Study

Annotation of protein sequences are important for the preservation and reuse of knowledge and for content-based queries. Traditional wet-lab methods are labor intensive and prone to human error. Alternatively, sequence-similarity-based tools are time intensive and require high investment in computing facilities. Therefore, a simple and practical method that is more accurate, faster, easy to configure and use, low computing cost, and exhaustive is needed. In this study, a GO-based tool named *extended* UTMGO is developed to meet these features. The tool is composed of two primary components. The first component named SMAGA is used to

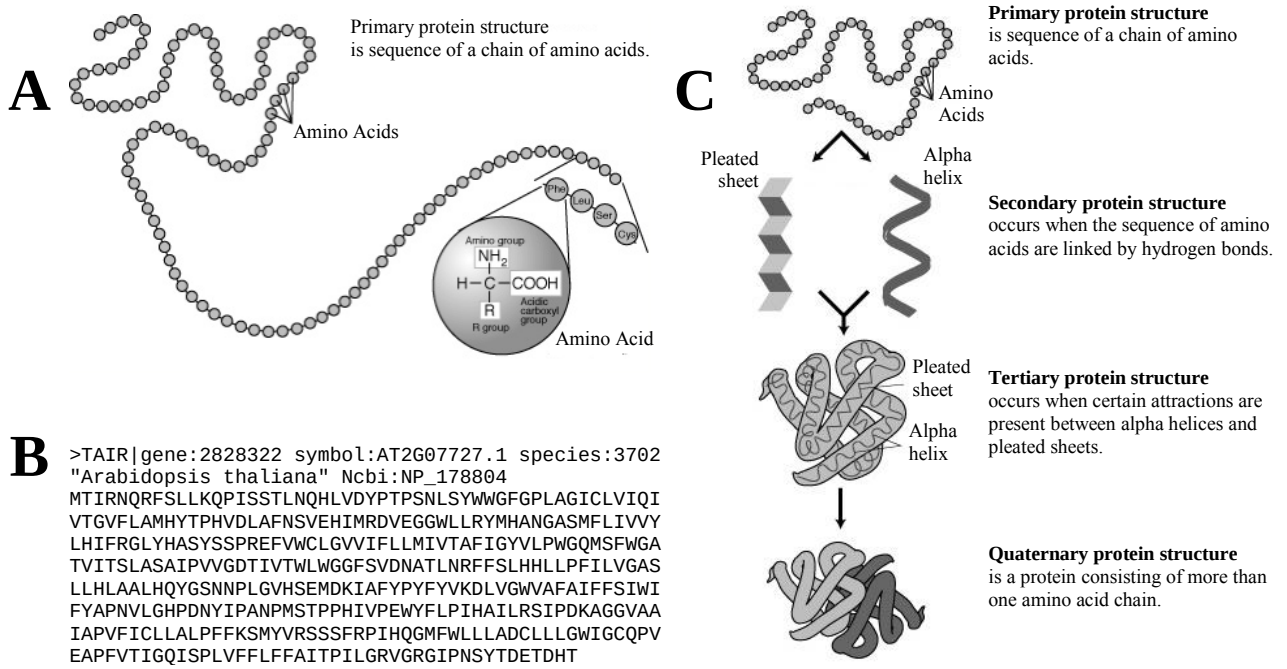

Fig. 1 The protein sequence illustration. (A) The protein primary structure (source: the National Human Genome Research Institute (NHGRI)). (B) The protein sequence of *AT2G07727.1* in FASTA format (source: The Arabidopsis Information Resource (TAIR)). (C) The four levels of protein structure (source: the National Human Genome Research Institute (NHGRI)).

split the monolithic GO RDF/XML file. The SMAGA applies parallel genetic algorithm and split-and-merge algorithm. The split-and-merge algorithm is implemented to improve infeasible clusters in order to efficiently estimate the number *k* of clusters. The second component named SSMGA is used to search for semantically related GO terms from the fragmented GO RDF/XML files. The SSMGA applies parallel genetic algorithm and semantic similarity measure algorithm. The semantic similarity measure algorithm is implemented due to its ability to improve the precision and recall of information retrieval by identifying the relation between GO terms. This is acquired by computing the distance or the amount of information those GO terms share in common. Both components use the parallel genetic algorithm because of its capability of being adaptive, efficient, robust, and a global search method that is suitable to address a situation where the search space is large. Moreover, parallel genetic algorithm optimizes its objective function by utilizing the genetic operators to find an optimal solution. It can also be executed on a low-cost PC cluster using message passing interface libraries that are open source and easy to install.

II. Protein Sequence Annotation

A protein sequence is a chain of amino acids that represents the primary structure of a protein as shown in Fig. 1. The protein sequence plays a central role to determine the structure, homology, and function of a protein as depicted in Fig. 2.

The database of protein sequences can be considered as primary database. It serves as a source for the construction of secondary databases that contain the results of analysis of the protein sequences in the primary databases. The secondary databases are related to protein families, domains, and functional sites. Examples of secondary databases are:

1) PROSITE (http://www.expasy.ch/prosite/) is a database of protein families, domains, and functional sites. The PROSITE is provided by the ExPASy (Expert Protein Analysis System) proteomics server of the Swiss Institute of Bioinformatics (SIB).
2) Pfam (http://www.sanger.ac.uk/Pfam/) comprises many common protein families and domains. It is a database managed by the Wellcome Trust Sanger Institute.
3) PANDIT (Protein and Associated Nucleotide Domains with Inferred Trees; http://www.ebi.ac.uk/goldman-srv/pandit/) is a protein families database developed and maintained by the European Bioinformatics Institute (EBI).

Recently, many works have used the protein sequence databases as main resource to predict protein-protein interactions [1], metabolic pathway [2], and protein subcellular localization [3].

The protein sequence databases are divided into two categories: the protein sequence repositories and the annotated protein sequence databases. The discussions of protein sequence databases have been presented by Whitfield *et al*. [4], Brooksbank *et al*. [5], and Apweiler *et al*. [6]. The protein sequence repositories are highly redundant and with little or no additional information to aid further analysis of the records. Among protein sequence repositories are NCBI (National Center for Biotechnology Information) Entrez

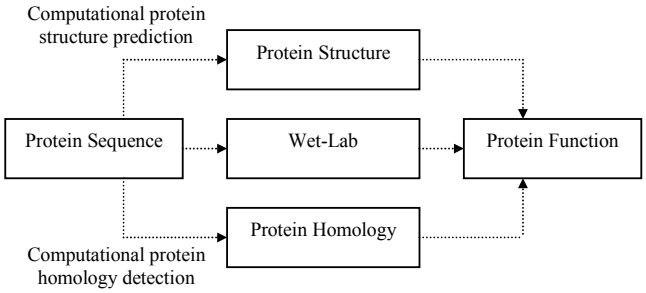

Fig. 2 Three different ways of inferring protein function from the protein sequence

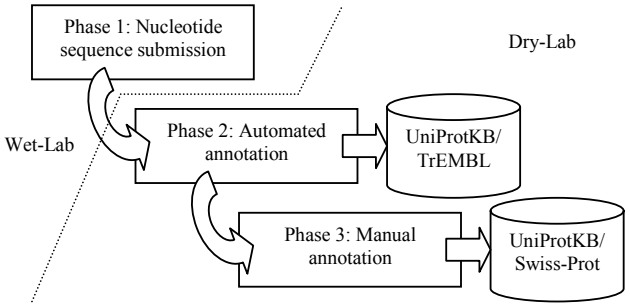

Fig. 3 Phases of protein sequence annotation in the UniProt

Protein (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=Protein) and RefSeq (Reference Sequence; http://www.ncbi.nlm.nih.gov/RefSeq/). On the other hand, the annotated protein sequence databases provide non-redundant set of protein sequences by consolidating all reports for a given protein sequence into one unique record. The annotation is done either manually by several expert biologists, automatically using bioinformatics tools like BLAST, or both combinations. By supplementing additional information to a protein sequence, it increases the value of the resource for users and can be regarded to be highly reliable. The most comprehensive annotated protein sequence database is UniProt (Universal Protein Resource; http://www.ebi.uniprot.org). The UniProt merges the information contained in UniProtKB/Swiss-Prot (Swiss Protein; http://www.ebi.ac.uk/swissprot/), UniProtKB/TrEMBL (Translated European Molecular Biology Laboratory; http://www.ebi.ac.uk/trembl/), and PIR (Protein Information Resource; http://pir.georgetown.edu/). The aim is to provide a central resource on protein sequences and functional annotation. The UniProt consists of three main components:

1) UniProtKB (UniProt Knowledgebase) provides extensive cross-references, functional and feature annotations, and literature-based evidence attribution for easy analysis and cross-database search. It comprises the manually annotated UniProtKB/Swiss-Prot section and the automatically annotated UniProtKB/TrEMBL section.
2) UniRef (UniProt Reference Clusters) offers speed similarity searches through sequence space compression by combining closely correlated sequences into a single record.
3) UniParc (UniProt Archive) stores all publicly available protein sequences, including their history and links to the source databases.

The UniProt is maintained collaboratively by the SIB and the EBI. Other annotated protein sequence databases are EXProt (Experimentally Verified Protein Functions; http://www.cmbi.kun.nl/EXProt/), PRF (Protein Research Foundation; http://www.prf.or.jp/en/), and TCDB (Transporter Classification Database; http://www.tcdb.org/).

The most systematic annotation of protein sequence is carried out by the UniProt. The protein sequences in the UniProt undergo three major phases of annotation as shown in Fig. 3. The process starts when the wet-lab researchers submit their nucleotide sequence to the EMBL. A similarity analysis including search for protein domains and the coding sequence (CDS) expected should be determined by the wet-lab researcher. Secondly, the CDS is translated into protein sequence. The protein sequence is then annotated automatically and stored in the UniProtKB/TrEMBL. The automated annotation is performed using automatically generated rules as in Spearmint [7] or manually curated rules based on protein families, including PIRSF classification-based name rules and site rules [8], HAMAP family rules [9], and RuleBase rules [10]. The UniProtKB/TrEMBL also received nucleotide sequences from GenBank (http://www.ncbi.nlm.nih.gov/Genbank/) and DDBJ (DNA Data Bank of Japan; http://www.ddbj.nig.ac.jp/) and protein sequences extracted from the literature or directly sent to the UniProtKB/Swiss-Prot. Thirdly, protein sequences in the UniProtKB/TrEMBL are selected for full manual annotation and consolidation into the UniProtKB/Swiss-Prot. The manual annotation is done by biologists and is based on literature curation and sequence analysis. The manual annotation procedures were described in detail by Apweiler *et al*. [11]. Further explanation of the annotation processes in the UniProt can be found in [12], [13].

Lately numerous methods have been proposed for automated protein sequence annotation. These methods can essentially be divided into four main classes as follows:

1) Sequence-similarity-based method depends on the determination of a local or global similarity between the not-yet annotated protein sequence and protein sequences with known annotation. This method uses sequence similarity search algorithms such as Smith-Waterman and Needleman-Wunsch algorithms. Examples of works have been carried out by Snyder *et al*. [14] and Koski *et al*. [15].
2) Controlled-vocabulary-based method employs the most widely used biological ontology, the GO along with its annotation databases to annotate protein sequence such as studies done by Jones *et al*. [16] and Prlic *et al*. [17].
3) Literature-based method relies on natural language processing and text mining techniques to extract information from the biomedical literature as evidence to annotate protein sequence. Some recent studies have been

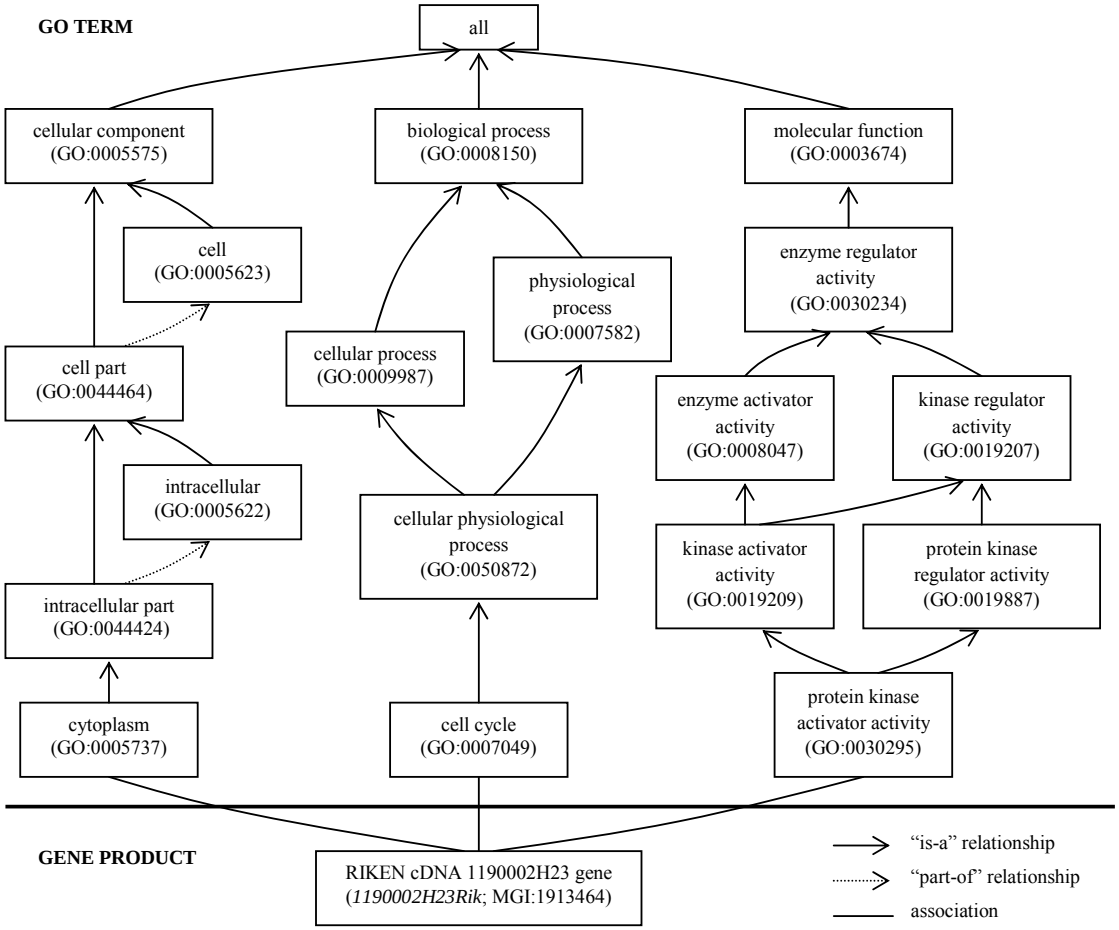

Fig. 4 The Gene Ontology sub-ontologies

conducted by Yuan *et al*. [18] and Chiang and Yu [19].

4) Rule-based method annotates protein sequence based on condition and existence of certain rules. The rules are created according to information extracted from the secondary databases. This method has been applied by Sigrist *et al*. [20] and Yu [21].

III. Gene Ontology

The GO project started in 1998 by collaboration between three model organism databases: FlyBase (http:// flybase.bio.indiana.edu/), SGD (Saccharomyces Genome Database; http://www.yeastgenome.org/), and MGI (Mouse Genome Informatics; http://www.informatics.jax.org/). Currently, databases participated in the GO project covers model organisms like *Arabidopsis thaliana, Caenorhabditis elegans*, *Danio rerio*, *Dictyostelium discoideum*, *Oryza*, *Rattus norvegicus*, and several protozoan parasites including *Leishmania major*, *Plasmodium falciparum*, and *Trypanosoma brucei*. The GO project is developed and maintained by the GO Consortium. The GO Consortium is currently formed by 16 entities such as EBI, University of Cambridge, University of California Berkeley, The Jackson Laboratory, Stanford University, and Princeton University. The GO is one of the ontologies that take part in the Open Biomedical Ontologies (OBO; http://obo.sourceforge.net/). The OBO is an umbrella project providing well-structured controlled vocabularies that are freely available and can be used across different biological and medical domains.

The goal of the GO project is to construct a well defined and standardized vocabulary for describing the roles of genes and gene products in any organism, even if the cell is evolving and their roles in the cells are changing. The purposes of producing the controlled vocabularies are to manage different names for the same concepts existing in various species, to support cross-species comparison and cross-databases search, and to assist annotation of vast amounts of biological data held in genome and protein databases. The main concept used in the development of the GO is ontology. The ontology is an explicit description of a domain. The ontology is created to define common vocabulary and to share common understanding of the meaning of any vocabulary used. The ontology has been developed in many fields such as chemical process engineering [22], ecoinformatics [23], and multimedia [24]. The ontology has also been implemented to solve various problems related to semantic web search [25], verification of conceptual models [26], and database

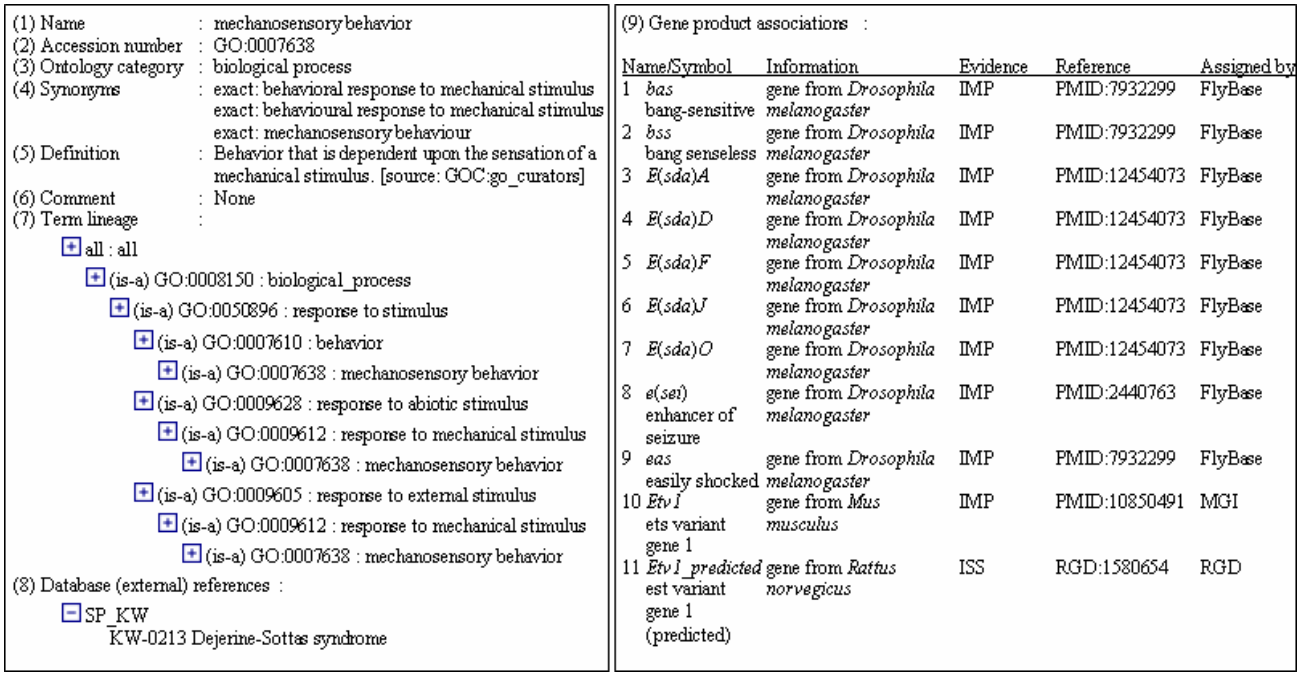

Fig. 5 The properties of Gene Ontology term. For example, the "mechanosensory behavior" (GO:0007638) and part of its gene product associations.

integration [27].

The GO comprises three sub-ontologies as shown in Fig. 4. The cellular component describes locations that refer to the place in the cell where a gene product is active like "cytoplasm" (GO:0005737). The biological process describes biological goals contributed by the gene or gene product such as "cell cycle" (GO:0007049). Finally, the molecular function describes activity of a gene product at the molecular level, an example includes "protein kinase activator activity" (GO:0030295). The vocabulary of the GO is called term. Each GO term is related to its parent either via: an "is-a" relationship like "intracellular part" (GO:0044424) is a "cell part" (GO:0044464); or a "part-of" relationship such as "intracellular part" (GO:0044424) is part of "intracellular" (GO:0005622). The properties of the GO term are depicted in Fig. 5. Each gene product associated to the GO term is supported by evidence code and specific reference. For example, an association between gene product "easily shock" (*eas*; FBgn0000536) and GO term "mechanosensory behavior" (GO:0007638) is supported by evidence code IMP (Inferred from Mutant Phenotype) and literature reference PMID:7932299 from PubMed (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?DB=pubmed). The evidence codes and its description can be found at http://www.geneontology.org/GO.evidence.shtml. The association of gene products to the GO terms is provided by Gene Ontology Annotation (GOA; http://www.ebi.ac.uk/GOA/). The GOA had successfully annotated proteins in the UniProtKB from a variety of species to the GO terms [28].

The GO data is stored in the following database categories:

1) *termdb* is a database that contains information on the GO terms and relationships only.
2) *assocdb* is a database which subsumes data in the *termdb* and addition with associations between the GO terms and gene products.
3) *seqdb* is a database containing protein sequences that associate with gene products and all data in the *assocdb*.
4) *seqdblite* is a database which is same as *seqdb*, except all IEA evidence associations have been taken out.

The GO data is in OBO, OWL, RDF/XML, and MySQL formats. The OBO and OWL formats are available just on the *termdb*. The MySQL format can be downloaded on all database categories. Meanwhile, the RDF/XML format comes without protein sequences and IEA evidence associations.

The GO has been used in many applications including gene expression studies [29], proteomics studies [30], comparative genomics [31], and data and text mining [32]. This is due to characteristics of the GO that the data is continuously evolved and refined, the structure is simple and relatively easy to understand and use, direct input from the biological community, and active curation to sustain the quality and integrity of data. Detail discussion about GO can be found in [33]–[39].

IV. Review of Algorithms

A. Splitting the Monolithic GO RDF/XML File

The GO RDF/XML is an XML document that is structured using various tags as a formalized way of describing the relationship between the GO terms. The RDF/XML syntax specification can be found at the World Wide Web Consortium (W3C) [40]. The GO RDF/XML is created to

allow the GO data to be shared and reused across the World Wide Web in a way that machines can interpret it. The goal of using this type of data storing is to prevent the user from manually importing the GO data into RDBMS format every time it is updated. Thus, the scientists with little computational background can avoid setting up the RDBMS software. The GO RDF/XML has been applied by numerous GO tools such as WEGO [41], a tool for plotting GO annotation results; ErmineJ [42], a tool for the analysis of gene sets in microarray gene expression data; DynGO [43], a tool to search for GO term and its association using batch and semantic retrieval; and COBrA [44], a browser and editor for GO and OBO ontologies that allows the user to make links between terms in those ontologies.

Due to large scale of the GO data as shown in Table 1 (as of November 2006), the GO RDF/XML is available without protein sequences and IEA evidence associations. But still the astronomical size and massive nature of this single flat file (current size is 446 MB) has caused the GO RDF/XML difficult to be maintained, published, validated, and processed. One way to make the GO RDF/XML more complete, coherent, and easy to browse is by splitting it into multiple files. Hence, it enables protein sequences and IEA evidence associations to be included in the GO RDF/XML.

Splitting the GO RDF/XML file requires the GO terms to be grouped into a number *k* of clusters. Since the GO terms are structured as a DAG, let GO graph G = {V, E} that consists of two main elements: V is a set of nodes that represent the GO terms and E is a set of edges that represent relationships between the GO terms. Partitioning the GO graph is a combinatorial problem and can be regarded as a Graph Partitioning Problem (GPP). The intention of GPP is to divide a vertex set V into *k* disjoint and non-empty subsets in order to produce partitions that have higher degree of interaction between nodes in the same partition and have lower degree of interaction between nodes in different partitions. The task of partitioning the large GO graph that contains more than 22 thousand nodes and almost 2.0 million paths is characterized as bearing very high computational complexity. Furthermore, identifying the number *k* of clusters is a hard algorithmic problem since it is difficult to guess, and it requires a trial-and-error work.

A large number of clustering algorithms have been proposed in the past decade. Among the successfully implemented algorithms are:

1) Fuzzy logic: e.g. fast generalized fuzzy c-means [45] for image segmentation and fuzzy-based cosine clustering [46] for anomaly detection in web documents.
2) Support vector machines: e.g. support vector clustering [47] for marketing segmentation and clustering support vector machines [48] for protein local structure prediction.
3) K-means: e.g. k-means range algorithm [49] for personalized data clustering in e-commerce and greedy elimination algorithm [50] for global gene trajectory clustering.
4) Evolutionary algorithms: e.g. hybrid-evolutionary-programming algorithms [51] for microbial growth studies and work done by Rogers and Kulkarni [52] for part types and machine types clustering in cellular manufacturing.

TABLE I
SIZE OF GENE ONTOLOGY

Item	No. of Records
GO terms	22,591
Definitions of GO terms	21,693
Synonyms for GO terms	20,517
Relationships between GO terms	34,367
All paths in GO graph	1,923,805
External database identifier entities	5,547,071
Links from GO terms to other databases	91,597
Gene products	2,320,059
Synonyms for gene products	315,857
Link between gene product and GO term	9,387,131
Gene product counts per GO term	541,680
Evidence type and reference for an association between gene product and GO term	10,679,104
External database links for an association between gene product and GO term	10,274,938
Protein sequences	2,122,707
Link between gene product and protein sequence	2,133,624
External database links for a protein sequence	19,727,005
Species	263,231

Other algorithms are: model-based clustering [53] for semiconductor fabrication process control; hidden Markov model based clustering [54] for analysis of gene expression time-course data; and self-organizing map [55] for segmentation of natural and synthetic diphthongs. There are also hybrid algorithms such as rough fuzzy c-means [56], rough k-means [57], and evolutionary fuzzy c-means [58]. Comparison of clustering algorithms can be found in [59]–[61].

For automatic clustering, several new algorithms have been developed recently, for examples:

1) Evolutionary clustering [62] employs merge and split mutation operators to dynamically change the number *k* of clusters that is represented by the length of the chromosome during the evolutionary process. This algorithm is specifically developed for gene expression microarray data analysis.
2) Laszlo and Mukherjee [63] introduces genetic algorithm for evolving centers in the k-means. They exploit the emersion of chromosomes with varying number of genes to simultaneously search for a range of good clusters around the specified *k*.
3) Hybrid niching genetic algorithm [64] uses Selecting Factor Group (SFG) and Comparing Factor Group (CFG). The SFG is used to encourage mating between chromosomes. Meanwhile, the purpose of the CFG is to balance competition during substitution between chromosomes with the same number of clusters and chromosomes with different number of clusters. Three real data sets of iris, breast cancer, and subcellcycle are

used in the experiments.

4) Simulated annealing using reversible jump Markov chain Monte Carlo [65] can automatically determine the correct number of clusters using various moves: birth move, death move, split move, merge move, and perturb move. The effectiveness of this algorithm has been demonstrated for automatically classifying the different land cover types in a satellite image.

In another part, GPP has been studied by the following researchers:

1) Aykanat *et al*. [66] has formulated adaptive object space decomposition problem as a GPP. A tool named RM-MeTiS is developed to partition the graph. This tool consists of three phases: multilevel coarsening, initial remapping, and multilevel refinement. The largest graph consists of 109,744 nodes and the experiments are conducted on a 28-node PC cluster.
2) Duarte *et al*. [67] has modeled image segmentation as a GPP. The GPP is resolved by a variant of normalized cut using hierarchical social metaheuristic. The experiments involve a graph with 11,155 nodes and 1,817,351 edges.
3) Boulif and Atif [68] has used GPP to deal with the manufacturing cell formation problem. A new branch-and-bound-enhanced genetic algorithm is proposed to solve the problem.
4) Mitchell and Mancoridis [69] has invented Bunch as a tool for modularization of software systems. This tool uses search techniques and treats the clustering process as a GPP. It has been applied to graphs with almost 10,000 nodes and 100,000 edges.

B. Searching for Semantically Similar GO Terms

Recently, the GO Bibliography (http://www.geneontology.org/cgi-bin/biblio.cgi), a listing of GO-related publications, has grown to over 1,100 articles. It documents a number of novel uses of the GO data and indicates the rapid progress of implementation of the GO terms to solve various bioinformatics problems. By contrast, the existing GO browsers to support basic needs for scientists to search the GO terms are still using conventional approach which is based on keyword matching. Thus, for a scientist to find a group of GO terms that have semantically similar properties is time consuming and a hard task. For example, as shown in Fig. 6, the keyword matching is not capable of computing the relationship between "intracellular organelle" (GO:0043229) and "cytoplasm" (GO:0005737). This is due to the fact that their names do not exactly or approximately match.

Semantic similarity search is required in order to search for semantically similar GO terms and to reduce dependency of specialists. Thence, it avoids the users from investing lots of time browsing the GO terms. However, this approach involves computing the amount of information the GO terms share in common and/or calculating the depth and the local network density of the GO term. This scenario becomes complicated since the GO terms are structured as a DAG and searching the GO graph is an NP-complete problem.

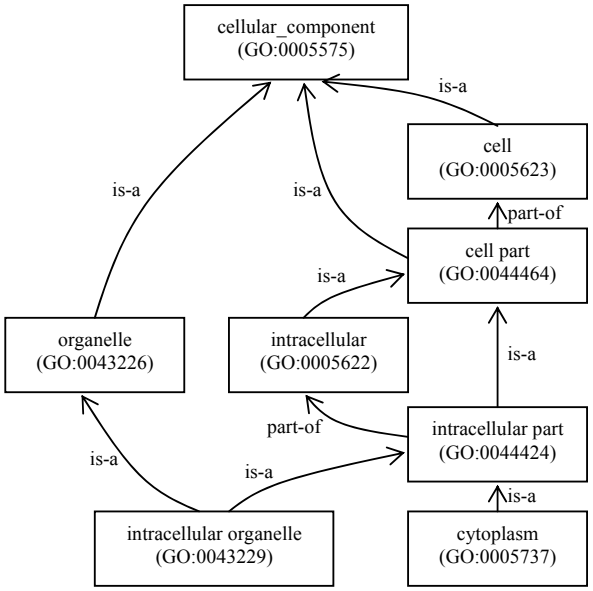

Fig. 6 An example of Gene Ontology terms

There are numerous search techniques that are frequently and extensively used in computer science, engineering, mathematics, and other fields such as:

1) Tabu search is a local search technique. It uses a local or neighborhood search procedure to repetitively move from a solution x to a solution x' in the neighborhood of x, until termination criterion is satisfied. Examples of application include flow shop problem [70] and facility location problem [71].
2) Simulated annealing is a global optimization technique that is based on probabilistic methods. It traverses the search space by producing neighboring solutions of the current solution. The simulated annealing has been applied in flexible manufacturing system [72] and heterogeneous distributed system [73].
3) Genetic algorithms are a global search heuristics. These algorithms work by seeking potential solutions and evaluating them. The best solutions are modified to form a new population. This operation is repeated until no better solutions are generated. The genetic algorithms have solved various problems such as nurse rerostering problem [74] and personnel assignment problem [75].
4) Ant colony optimization is a population-based technique that tries numerous solution options at each step of the algorithm. The ant colony optimization is inspired by the behavior of ants in discovering routes from the colony to food. It has been applied in water distribution system [76] and solved the nonlinear resource allocation problem [77].

Other techniques include particle swarm optimization [78], hill climbing [79], and cross-entropy method [80]. A detailed comparison among these techniques can be found in [81]–[83].

In the case of similarity search, researchers have used

different measures to identify similarity between two concepts being compared. Lately, several new similarity measures have been introduced such as:

1) Edge-similarity measure [84] is applied to varying image illumination and contrast.
2) Quantitative tract similarity measure [85] is based on the shape and length of the two tracts being analyzed to improve image segmentation reproducibility.
3) Trainable similarity measure [86] applied the matching-pursuit approach for road-sign classification.
4) Clip-based similarity measure [87] is based on two bipartite graph matching algorithms (maximum matching and optimal matching) for video retrieval and video summarization.
5) Spectral similarity measures [88] consist of four spectral measures (spectral angel measure, Euclidean distance measure, spectral correlation measure, and spectral information divergence) for the analysis of hyperspectral imagery.

Other similarity measures are: Popescu *et al*. [89] and Chen *et al*. [90] have proposed fuzzy similarity measure for gene product similarity and distorted fingerprints matching respectively; and Lee and Crawford [91] and Moghaddam *et al*. [92] have created Bayesian similarity measure for image segmentation and image matching respectively. Evaluation of different similarity measures have been done by Skerl *et al*. [93] for rigid registration of medical images and Núñez *et al*. [94] on improving case-based reasoning for environmental decision support systems.

A list of tools for searching and browsing the GO terms can be found at http://www.geneontology.org/GO.tools.browsers.shtml. All these tools are free to academics, among them are:

1) CGAP GO Browser is developed by The Cancer Genome Anatomy Project. It allows the user to browse the GO terms using the hierarchy view and find the known human and mouse genes assigned to each term. This tool can be used at http://cgap.nci.nih.gov/Genes/GOBrowser/.
2) GOFish is created using Java applet by the Roth Laboratory at the Harvard University. It uses term name or accession number as an input and then performs keyword matching. This tool allows the user to construct arbitrary Boolean queries using GO terms, and ranks gene products that satisfy the queries. The GOFish can be found at http://llama.med.harvard.edu/software.html.
3) Ontology Lookup Service is provided by the European Bioinformatics Institute. It is based on partial keyword search. As the users types into the search box, they will see recommended terms that match what are being entered in the list box. This tool was developed to merge all publicly available biomedical ontologies into a single database. It can be viewed at http://www.ebi.ac.uk/ontology-lookup/.

Other browsers are AmiGO (http://godatabase.org/), EP GO Browser (http://ep.ebi.ac.uk/EP/GO/), QuickGO Browser (http://www.ebi.ac.uk/ego/), GenNav Browser (http://mor.nlm.nih.gov/perl/gennav.pl), and MGI GO Browser (http://www.informatics.jax.org/searches/GO_form.shtml).

C. References

Bioinformatics is the application of computer technology to store, retrieve, analyze, simulate, or predict the composition or the structure of biomolecules. It involves the development of algorithms and statistical techniques, databases, and tools. The bioinformatics tools should be developed using open source and web technologies. Therefore, these tools can be distributed freely and used extensively by the bioscientists. However, an excellent tool should be easy to be setup and used, can be run on low-cost hardware, and requires a short execution time.

Recently, a number of bioinformatics tools have been developed for annotation of protein sequence based on the GO data. These tools are:

1) Blast2GO employs BLAST to find homologous sequences to FASTA (Fast Alignment) formatted input protein sequences. The Blast2GO extracts the GO terms for each found hit by mapping to existing annotation associations. An annotation rule finally assigns GO terms to the query protein sequence. This tool can be accessed at http://bioinfo.ivia.es/blast2go/. It is maintained by the Centro de Genómica at the Instituto Valenciano de Investigaciones Agrarias.
2) GoAnna can be applied for protein sequence annotation using a sequence similarity search. This tool accepts a list of protein sequences in FASTA format. The GoAnna conducts BLAST search against AgBase databases or GO annotated databases like UniProtKB/Swiss-Prot and UniProtKB/TrEMBL. This tool is developed by the Mississippi State University and can be used at http://agbase.msstate.edu/GOAnna.html.
3) HT-GO-FAT provides the bioscientists with a high-throughput mapping of unknown protein sequence to GO annotation. It uses BLAST for sequence similarity search. The HT-GO-FAT can be downloaded from http://liru.ars.usda.gov/mainbioinformatics.html. This tool is developed by the Livestock Issues Research Unit at the USDA Agricultural Research Service.
4) InGOt is capable to assign up-to-date GO terms to a given protein sequence. The InGOt claims to have more sequences than any public resource and assignments harvested from the broadest possible GO-linked resources. It is proprietary software by Inpharmatica Ltd. A free two week trial of this tool can be downloaded at http://www.inpharmatica.co.uk/ingot/.

Other GO-based protein sequence annotation tools are: GOPET is addressable via http://genius.embnet.dkfz-heidelberg.de/menu/biounit/open-husar/, and it has been developed by the German Cancer Research Center; GOtcha (http://www.compbio.dundee.ac.uk/gotcha/gotcha.php) by the Barton Group at the University of Dundee; GoFigure (http://udgenome.ags.udel.edu/gofigure/) is under the UDGenome project by the University of Delaware; GOblet (http://goblet.molgen.mpg.de/) is introduced by the Max Planck Institute for

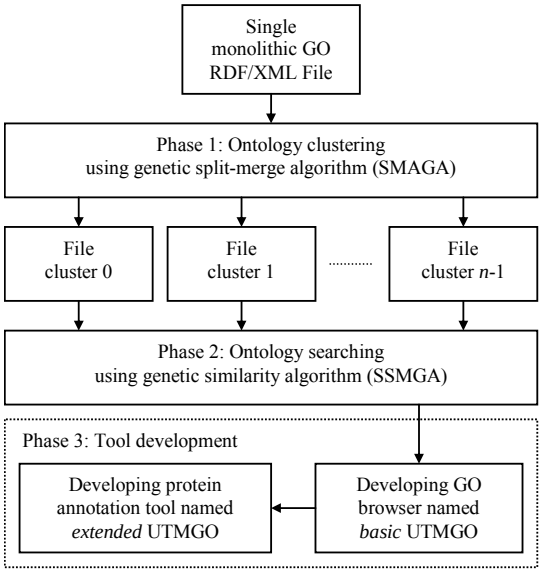

Fig. 7 The proposed framework

Molecular Genetics; and lastly JAFA (http://jafa.burnham.org/) is maintained by the Burnham Institute for Medical Research.

In parallel, several works using computational intelligence techniques for annotation of protein sequence have also been done by:

1) Kirac *et al*. [95] introduced a data mining technique that calculates the probabilistic relationships between the GO annotations of proteins on protein-protein interaction data. Then, it assigns highly associated GO terms of annotated proteins to the target protein sequence.
2) Ray and Craven [96] built a system to annotate a given protein sequence with codes from the GO using the text of an article from the biomedical literature as evidence. This system relies on statistical techniques namely the *n*-gram models and the Naïve Bayes models.
3) Ponomarenko *et al*. [97] shows how protein sequence annotation can be improved and corrected if protein structures are available. They used the combinatorial extension algorithm to compare the structure. Then, it widens the protein annotation provided by the GOA to further annotate the protein sequences in the PDB (Protein Data Bank; http://www.rcsb.org/pdb/).

There are also varieties of protein sequence annotation tools that have been developed without depending on the GO data such as ProtoBee (http://www.protobee.cs.huji.ac.il/), KOBAS (http://kobas.cbi.pku.edu.cn/), MineBlast (http://leger2.gbf.de/cgi-bin/MineBlast.pl), ProFAT (http://cluster-1.mpi-cbg.de/profat/), and FeatureMap3D (http://www.cbs.dtu.dk/services/FeatureMap3D/).

V. Methodology

A. The Proposed Framework and Results

The proposed framework involved three main phases

```
1   Algorithm SMAGA (G, k_min);
2   Input: G = {V, E} (a Gene Ontology Graph) and k_min (a minimum
3   number of clusters)
4   Output: C = (C_1, C_2, …, C_k) (a clustering)
5   begin
6     t := 0;
7     initialize P(t);
8     evaluate P(t);
9     while not termination-condition do
10      t := t + 1;
11      select P(t) from P(t – 1);
12      alter P(t) by crossover and mutation operator;
13      alter P(t) by split and merge function;
14      evaluate P(t);
15    end-while
16  end
```

Fig. 8 The SMAGA algorithm

namely the ontology clustering phase, the ontology searching phase, and the tool development phase as depicted in Fig. 7.

In the first phase, the SMAGA is formed to cluster the GO terms. The aim is to split the single monolithic GO RDF/XML file into a number of smaller files. The SMAGA is a combination of split-and-merge algorithm and parallel genetic algorithm. The detail about SMAGA is discussed by Othman *et al*. [98]. The overview of the SMAGA algorithm is shown in Fig. 8. The SMAGA is capable of automatically identifying the number *k* of clusters, producing balanced clusters in terms of number of elements in each cluster, and demands reasonable amount of processing time.

In the second phase, the SSMGA is constructed to perform similarity search. The idea is to find a group of semantically similar GO terms for a given query term. The SSMGA incorporates semantic similarity measure algorithm in the parallel genetic algorithm. A comprehensive discussion of the SSMGA is done by Othman *et al*. [99]. The SSMGA algorithm consists of the following steps:

1) Perform preprocessing using the semantic similarity measure algorithm.
2) Initialization of a population of chromosomes where alleles for each chromosome show either the GO terms are retrieved or not retrieved.
3) Evaluate the fitness of each chromosome.
4) Select chromosomes for reproduction using the roulette wheel selection scheme.
5) Apply two-point crossover and swap mutation operators.
6) Replace the least fit chromosomes in the existing population by the newly generated offspring.
7) Repeat steps (3)–(6) until the stopping criteria are met.

The inputs for the SSMGA algorithm are the GO graph and the query GO term. This algorithm returns the best chromosome representing a set of GO terms that are semantically similar to the query term. The SSMGA is susceptible of returning the GO terms that do not contain the keyword specified by the user. Furthermore, it is able to avoid producing many GO terms with low similarity score and can be executed in a short time.

In the third phase, the *basic* UTMGO is developed using web technology. The main goal of this tool is to act as a new way to search the GO terms. The *basic* UTMGO has shown its

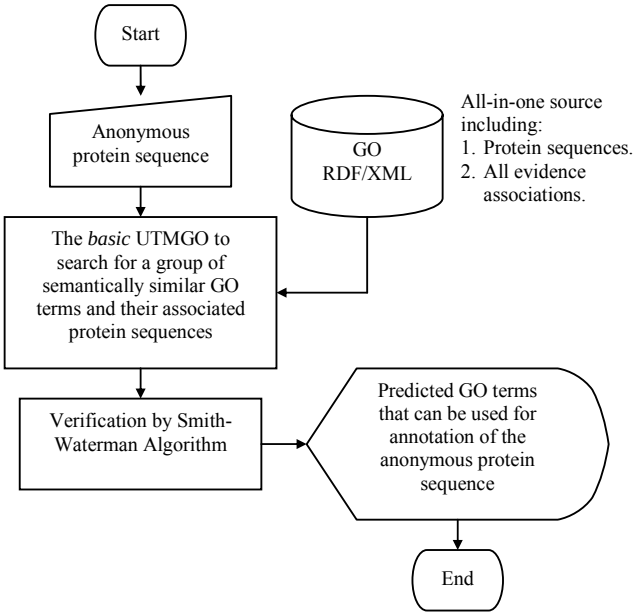

Fig. 9 The flowchart of the *extended* UTMGO

capability to determine the semantically similar GO terms as compared to other keyword-based GO browsers. This is due to the effectiveness of the SMAGA and SSMGA because of its intelligent components. The potential of this tool has been broadened to annotate protein sequences. The tool named *extended* UTMGO is able to return a set of GO terms together with their associated protein sequences that have higher sequence alignment score to the query protein sequence. This feature allows bioscientists to annotate anonymous protein sequences by only using the GO terms. Thus, it prevents dependency on BLAST and blind sequence similarity search. Both of these tools are described by Othman *et al*. [100]. The flowchart of the *extended* UTMGO is shown in Fig. 9.

B. Data Sources

The GO data used in this study is in RDF/XML format. The data is compressed in a GZIP file named go_YYYYMM-assocdb.rdf-xml.gz. The data is updated monthly and can be downloaded from http://archive.godatabase.org/. The data comes without protein sequences and IEA evidence associations. Therefore, to include both of them into the GO RDF/XML file these data are taken from the MySQL format. The GO data in MySQL format is stored in a file named go_YYYYMM-seqdb-tables.tar.gz.

In the meantime, to assess the performance of the *extended* UTMGO for annotating protein sequences, 50 protein sequences are selected randomly from each species as follows:

1) *Oryza sativa ssp japonica* from the Gramene database (http://www.gramene.org/Oryza_sativa/index.html).
2) *Homo sapiens* is obtained from the Ensembl database (http://www.ensembl.org/Homo_sapiens/index.html).
3) *Saccharomyces cerevisiae* from the SGD database.
4) *Arabidopsis thaliana* is downloaded from the TAIR database (The Arabidopsis Information Resource; http://www.arabidopsis.org/).

C. Instrumentation and Comparison

All experiments are run on a 25-node low-cost PC cluster with 2.8GHz Pentium IV of processor, 512MB of memory, and 100Mbps of network speed. The operating system used is Fedora Core 5. The low-cost PC cluster is based on island (coarse-grained) model that has been successfully used for parallel genetic algorithms [101]–[103]. It is implemented using MPICH2 libraries (http://www-unix.mcs.anl.gov/mpi/mpich/) created by the Argonne National Laboratory. The genetic algorithms used in this study are an enhancement of the existing GAlib C++ libraries (http://lancet.mit.edu/ga/). The interface for the *basic* and *extended* UTMGO are developed using Java Server Pages (JSP) scripts.

The performance of the proposed tools is measured by recall, precision, and running time of the results. The *basic* UTMGO has been compared with AmiGO, GenNav Browser, TAIR Keyword Browser, and QuickGO Browser. The term similarity score of the GO terms returned by each browser is also presented. On the other hand, the capability of the *extended* UTMGO is compared with other GO-based protein sequence annotation tools such as GOPET, GOtcha, GoFigure, and JAFA. This study has also presented the sequence alignment score of the protein sequences associated to the GO terms that are returned by each protein sequence annotation tool.

VI. Summary

The aim of this paper is to give an overview of the weaknesses of existing protein sequence annotation tools, thus, to introduce a new tool named *extended* UTMGO that is fully based on the GO to overcome those weaknesses. However, to come out with the *extended* UTMGO, two main problems that are related to automatic clustering and semantic similarity search have been raised and their solutions have been discussed. The automatic clustering has been solved using combination of split-and-merge algorithm and parallel genetic algorithm. On the other hand, the semantic similarity measure algorithm has been incorporated in the parallel genetic algorithm to perform the semantic similarity search. The review on automatic clustering and semantic similarity search algorithms including their applications has been presented to support the justification of the chosen algorithms. This paper also gives broad review of basic concepts of the protein sequence, protein sequence databases, and processes involved in the protein sequence annotation for better understanding of the nature of the problems, together with explanation about GO including its properties, characteristics, and applications.

References

[1] A. Chinnasamy, A. Mittal, and W.K. Sung, "Probabilistic prediction of protein-protein interactions from the protein sequences," *Computers in Biology & Medicine*, vol. 36, no. 10, pp. 1143-1154, Oct. 2006.

[2] L. Pireddu, D. Szafron, P. Lu, and R. Greiner, "The Path-A metabolic pathway prediction web server," *Nucleic Acids Research*, vol. 34, no. 1, pp. W714-W719, Jul. 2006.
[3] G.K. Acquaah-Mensah, S.M. Leach, and C. Guda, "Predicting the subcellular localization of human proteins using machine learning and exploratory data analysis," *Genomics Proteomics Bioinformatics*, vol. 4, no. 2, pp. 120-133, May 2006.
[4] E.J. Whitfield, M. Pruess, and R. Apweiler, "Bioinformatics database infrastructure for biotechnology research," *J. Biotechnology*, vol. 124, no. 4, pp. 629-639, Aug. 2006.
[5] C. Brooksbank, G. Cameron, and J. Thornton, "The European Bioinformatics Institute's data resources: towards systems biology," *Nucleic Acids Research*, vol. 33, no. 1, pp. D46-D53, Jan. 2005.
[6] R. Apweiler, A. Bairoch, and C.H. Wu, "Protein sequence databases," *Current Opinion in Chemical Biology*, vol. 8, no. 1, pp. 76-80, Feb. 2004.
[7] E. Kretschmann, W. Fleischmann, and R. Apweiler, "Automatic rule generation for protein annotation with the C4.5 data mining algorithm applied on SWISS-PROT," *Bioinformatics*, vol. 17, no. 10, pp. 920-926, Oct. 2001.
[8] C.H. Wu, H. Huang, L.S. Yeh, and W.C. Barker, "Protein family classification and functional annotation," *Computational Biology & Chemistry*, vol. 27, no. 1, pp. 37-47, Feb. 2003.
[9] A. Gattiker, K. Michoud, C. Rivoire, A.H. Auchincloss, E. Coudert, T. Lima, P. Kersey, M. Pagni, C.J. Sigrist, C. Lachaize, A.L. Veuthey, E. Gasteiger, and A. Bairoch, "Automated annotation of microbial proteomes in SWISS-PROT," *Computational Biology & Chemistry*, vol. 27, no. 1, pp. 49-58, Feb. 2003.
[10] W. Fleischmann, S. Moller, A. Gateau, and R. Apweiler, "A novel method for automatic functional annotation of proteins," *Bioinformatics*, vol. 15, no. 3, pp. 228-233, Mar. 1999.
[11] R. Apweiler, A. Bairoch, C.H. Wu, W.C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, H. Huang, R. Lopez, M. Magrane, M.J. Martin, D.A. Natale, C. O'Donovan, N. Redaschi, and L.S. Yeh, "UniProt: the Universal Protein knowledgebase," *Nucleic Acids Research*, vol. 32, no. 1, pp. D115-D119, Jan. 2004.
[12] C.H. Wu, R. Apweiler, A. Bairoch, D.A. Natale, W.C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, H. Huang, R. Lopez, M. Magrane, M.J. Martin, R. Mazumder, C. O'Donovan, N. Redaschi, and B. Suzek, "The Universal Protein Resource (UniProt): an expanding universe of protein information," *Nucleic Acids Research*, vol. 34, no. 1, pp. D187-D191, Jan. 2006.
[13] A. Bairoch, R. Apweiler, C.H. Wu, W.C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, H. Huang, R. Lopez, M. Magrane, M.J. Martin, D.A. Natale, C. O'Donovan, N. Redaschi, and L.S. Yeh, "The Universal Protein Resource (UniProt)," *Nucleic Acids Research*, vol. 33, no. 1, pp. D154-D159, Jan. 2005.
[14] K.A. Snyder, H.J. Feldman, M. Dumontier, J.J. Salama, and C.W. Hogue, "Domain-based small molecule binding site annotation," *Bioinformatics*, vol. 22, no. 3, pp. 291-296, Feb. 2006.
[15] L.B. Koski, M.W. Gray, B.F. Lang, and G. Burger, "AutoFACT: an automatic functional annotation and classification tool," *BMC Bioinformatics*, vol. 6, no. 1, rec. 151, Jun. 2005.
[16] C.E. Jones, U. Baumann, and A.L. Brown, "Automated methods of predicting the function of biological sequences using GO and BLAST," *BMC Bioinformatics*, vol. 6, no. 1, rec. 272, Nov. 2005.
[17] A. Prlic, F.S. Domingues, P. Lackner, and M.J. Sippl, "WILMA-automated annotation of protein sequences," *Bioinformatics*, vol. 20, no. 1, pp. 127-128, Jan. 2004.
[18] X. Yuan, Z.Z. Hu, H.T. Wu, M. Torii, M. Narayanaswamy, K.E. Ravikumar, K. Vijay-Shanker, and C.H. Wu, "An online literature mining tool for protein phosphorylation," *Bioinformatics*, vol. 22, no. 13, pp. 1668-1669, Jul. 2006.
[19] J.H. Chiang and H.C. Yu, "Literature extraction of protein functions using sentence pattern mining," *IEEE Trans. Knowledge & Data Engineering*, vol. 17, no. 8, pp. 1088-1098, Aug. 2005.
[20] C.J.A. Sigrist, E.D. Castro, P.S. Langendijk-Genevaux, V.L. Saux, A. Bairoch, and N. Hulo, "ProRule: a new database containing functional and structural information on PROSITE profiles," *Bioinformatics*, vol. 21, no. 21, pp. 4060-4066, Aug. 2005.
[21] G.X. Yu, "Ruleminer: a knowledge system for supporting high-throughput protein function annotations," *J. Bioinformatics & Computational Biology*, vol. 2, no. 4, pp. 595-617, Dec. 2004.
[22] J. Morbach, A. Yang, and W. Marquardt, "OntoCAPE—a large-scale ontology for chemical process engineering," *Engineering Applications Artificial Intelligence*, vol. 20, no. 2, pp. 147-161, Mar. 2007.
[23] R.J. Williams, N.D. Martinez, and J. Golbeck, "Ontologies for ecoinformatics," *Web Semantics: Science, Services & Agents on the World Wide Web*, vol. 4, no. 4, pp. 237-242, Dec. 2006.
[24] M. Naphade, J.R. Smith, J. Tesic, S. Chang, W. Hsu, L. Kennedy, A. Hauptmann, and J. Curtis, "Large-scale concept ontology for multimedia," *IEEE Multimedia*, vol. 13, no. 3, pp. 86-91, Jul.-Sep. 2006.
[25] J. Köhler, S. Philippi, M. Specht, and A. Rüegg, "Ontology based text indexing and querying for the semantic web," *Knowledge-Based Systems*, vol. 19, no. 8, pp. 744-754, Dec. 2006.
[26] C. Hess and C. Schlieder, "Ontology-based verification of core model conformity in conceptual modeling," *Computers, Environment & Urban Systems*, vol. 30, no. 5, pp. 543-561, Sep. 2006.
[27] D. Pérez-Rey, V. Maojo, M. García-Remesal, R. Alonso-Calvo, H. Billhardt, F. Martin-Sánchez, and A. Sousa, "ONTOFUSION: ontology-based integration of genomic and clinical databases," *Computers in Biology & Medicine*, vol. 36, no. 7-8, pp. 712-730, Jul.-Aug. 2006.
[28] E. Camon, M. Magrane, D. Barrell, V. Lee, E. Dimmer, J. Maslen, D. Binns, N. Harte, R. Lopez, and R. Apweiler, "The Gene Ontology Annotation (GOA) database: sharing knowledge in Uniprot with gene ontology," *Nucleic Acids Research*, vol. 32, no. 1, pp. D262-266, Jan. 2004.
[29] A. Lewin and I.C. Grieve, "Grouping gene ontology terms to improve the assessment of gene set enrichment in microarray data," *BMC Bioinformatics*, vol. 7, no. 1, rec. 426, Oct. 2006.
[30] X. Wu, L. Zhu, J. Guo, D.Y. Zhang, and K. Lin, "Prediction of yeast protein-protein interaction network: insights from the gene ontology and annotations," *Nucleic Acids Research*, vol. 34, no. 7, pp. 2137-2150, Apr. 2006.
[31] Z. Cai, X. Mao, S. Li, and L. Wei, "Genome comparison using Gene Ontology (GO) with statistical testing," *BMC Bioinformatics*, vol. 7, no. 1, rec. 374, Aug. 2006.
[32] B. Zheng, D.C. McLean, and X. Lu, "Identifying biological concepts from a protein-related corpus with a probabilistic topic model," *BMC Bioinformatics*, vol. 7, no. 1, rec. 58, Feb. 2006.
[33] The Gene Ontology Consortium, "The Gene Ontology (GO) project in 2006," *Nucleic Acids Research*, vol. 34, no. 1, pp. D322-D326, Jan. 2006.
[34] J. Lomax, "Get ready to GO! A biologist's guide to the gene ontology," *Briefings in Bioinformatics*, vol. 6, no. 3, pp. 298-304, Sep. 2005.
[35] M.A. Harris, J. Lomax, A. Ireland, and J.I. Clark, "The gene ontology project," in *Encyclopedia Genetics, Genomics, Proteomics & Bioinformatics*, part 4, S. Subramaniam, Ed. New York: John Wiley & Sons, 2005.
[36] M. Bada, R. Stevens, C. Goble, Y. Gil, M. Ashburner, J.A. Blake, J.M. Cherry, M.A. Harris, and S. Lewis, "A short study on the success of the gene ontology," *J. Web Semantics*, vol. 1, no. 2, pp. 235-240, Feb. 2004.
[37] The Gene Ontology Consortium, "The Gene Ontology (GO) database and informatics resource," *Nucleic Acids Research*, vol. 32, no. 1, pp. D258-D261, Jan. 2004.
[38] The Gene Ontology Consortium, "Creating the gene ontology resource: design and implementation," *Genome Research*, vol. 11, no. 8, pp. 1425-1433, Aug. 2001.
[39] The Gene Ontology Consortium, "Gene ontology: tool for the unification of biology," *Nature Genetics*, vol. 25, no. 1, pp. 25-29, May 2000.
[40] D. Beckett. (2004, Feb. 10). RDF/XML syntax specification (revised) [Online]. Available: http://www.w3.org/TR/rdf-syntax-grammar.
[41] J. Ye, L. Fang, H. Zheng, Y. Zhang, J. Chen, Z. Zhang, J. Wang, S. Li, R. Li, L. Bolund, and J. Wang, "WEGO: a web tool for plotting GO annotations," *Nucleic Acids Research*, vol. 34, no. 1, pp. W293-W297, Jul. 2006.
[42] H.K. Lee, W. Braynen, K. Keshav, and P. Pavlidis, "ErmineJ: tool for functional analysis of gene expression data sets," *BMC Bioinformatics*, vol. 6, no. 1, rec. 269, Nov. 2005.
[43] H. Liu, Z.Z. Hu, and C.H. Wu, "DynGO: a tool for visualizing and mining of gene ontology and its associations," *BMC Bioinformatics*, vol. 6, no. 1, rec. 201, Aug. 2005.
[44] S. Aitken, R. Korf, B. Webber, and J. Bard, "COBrA: a bio-ontology editor," *Bioinformatics*, vol. 21, no. 6, pp. 825-826, Mar. 2005.

[45] W. Cai, S. Chen, and D. Zhang, "Fast and robust fuzzy c-means clustering algorithms incorporating local information for image segmentation," *Pattern Recognition*, vol. 40, no. 3, pp. 825-838, Mar. 2007.
[46] M. Friedman, M. Last, Y. Makover, and A. Kandel, "Anomaly detection in web documents using crisp and fuzzy-based cosine clustering methodology," *Information Sciences*, vol. 177, no. 2, pp. 467-475, Jan. 2007.
[47] J.J. Huang, G.H. Tzeng, and C.S. Ong, "Marketing segmentation using support vector clustering," *Expert Systems with Applications*, vol. 32, no. 2, pp. 313-317, Feb. 2007.
[48] W. Zhong, J. He, R. Harrison, P.C. Tai, and Y. Pan, "Clustering support vector machines for protein local structure prediction," *Expert Systems with Applications*, vol. 32, no. 2, pp. 518-526, Feb. 2007.
[49] G.P. Papamichail and D.P. Papamichail, "The k-means range algorithm for personalized data clustering in e-commerce," *European J. Operational Research,* vol. 177, no. 3, pp. 1400-1408, Mar. 2007.
[50] Z.S.H. Chan, L. Collins, and N. Kasabov, "An efficient greedy k-means algorithm for global gene trajectory clustering," *Expert Systems with Applications*, vol. 30, no. 1, pp. 137-141, Jan. 2006.
[51] A.C. Martinez-Estudillo, C. Hervas-Martinez, F.J. Martinez-Estudillo, and N. Garcia-Pedrajas, "Hybridization of evolutionary algorithms and local search by means of a clustering method," *IEEE Trans. Systems, Man & Cybernetics, Part B*, vol. 36, no. 3, pp. 534-545, Jun. 2006.
[52] D.F. Rogers and S.S. Kulkarni, "Optimal bivariate clustering and a genetic algorithm with an application in cellular manufacturing," *European J. Operational Research*, vol. 160, no. 2, pp. 423-444, Jan. 2005.
[53] J.Y. Hwang and W. Kuo, "Model-based clustering for integrated circuit yield enhancement," *European J. Operational Research*, vol. 178, no. 1, pp. 143-153, Apr. 2007.
[54] Y. Zeng and J. Garcia-Frias, "A novel HMM-based clustering algorithm for the analysis of gene expression time-course data," *Computational Statistics & Data Analysis*, vol. 50, no. 9, pp. 2472-2494, May 2006.
[55] H.M. Torres, J.A. Gurlekian, H.L. Rufiner, and M.E. Torres, "Self-organizing map clustering based on continuous multiresolution entropy," *Physica A: Statistical Mechanics & its Applications*, vol. 361, no. 1, pp. 337-354, Feb. 2006.
[56] S. Mitra, H. Banka, and W. Pedrycz, "Rough–fuzzy collaborative clustering," *IEEE Trans. Systems, Man & Cybernetics, Part B*, vol. 36, no. 4, pp. 795-805, Aug. 2006.
[57] G. Peters, "Some refinements of rough k-means clustering," *Pattern Recognition*, vol. 39, no. 8, pp. 1481-1491, Aug. 2006.
[58] E. Zio and P. Baraldi, "Evolutionary fuzzy clustering for the classification of transients in nuclear components," *Progress in Nuclear Energy*, vol. 46, no. 3-4, pp. 282-296, Apr. 2005.
[59] S.A. Mingoti and J.O. Lima, "Comparing SOM neural network with fuzzy c-means, k-means and traditional hierarchical clustering algorithms," *European J. Operational Research*, vol. 174, no. 3, pp. 1742-1759, Nov. 2006.
[60] G. Chicco, R. Napoli, and F. Piglione, "Comparisons among clustering techniques for electricity customer classification," *IEEE Trans. Power Systems*, vol. 21, no. 2, pp. 933-940, May 2006.
[61] H. Guldemır and A. Sengur, "Comparison of clustering algorithms for analog modulation classification," *Expert Systems with Applications*, vol. 30, no. 4, pp. 642-649, May 2006.
[62] P.C.H. Ma, K.C.C. Chan, X. Yao, and D.K.Y. Chiu, "An evolutionary clustering algorithm for gene expression microarray data analysis," *IEEE Trans. Evolutionary Computation*, vol. 10, no. 3, pp. 296-314, Jun. 2006.
[63] M. Laszlo and S. Mukherjee, "A genetic algorithm using hyper-quadtrees for low-dimensional k-means clustering," *IEEE Trans. Pattern Analysis & Machine Intelligence*, vol. 28, no. 4, pp. 533-543, Apr. 2006.
[64] W. Sheng, W. Swift, L. Zhang, and X. Liu, "A weighted sum validity function for clustering with a hybrid niching genetic algorithm," *IEEE Trans. Systems, Man & Cybernetics, Part B*, vol. 35, no. 6, pp. 1156-1167, Dec. 2005.
[65] S. Bandyopadhyay, "Simulated annealing using a reversible jump Markov chain Monte Carlo algorithm for fuzzy clustering," *IEEE Trans. Knowledge & Data Engineering*, vol. 17, no. 4, pp. 479-490, Apr. 2005.
[66] C. Aykanat, B.B. Cambazoglu, F. Findik, and T. Kurc, "Adaptive decomposition and remapping algorithms for object-space-parallel direct volume rendering of unstructured grids," *J. Parallel & Distributed Computing*, vol. 67, no. 1, pp. 77-99, Jan. 2007.
[67] A. Duarte, Á. Sánchez, F. Fernández, and A.S. Montemayor, "Improving image segmentation quality through effective region merging using a hierarchical social metaheuristic," *Pattern Recognition Letters*, vol. 27, no. 11, pp. 1239-1251, Aug. 2006.
[68] M. Boulif and K. Atif, "A new branch-&-bound-enhanced genetic algorithm for the manufacturing cell formation problem," *Computers & Operations Research*, vol. 33, no. 8, pp. 2219-2245, Aug. 2006.
[69] B.S. Mitchell and S. Mancoridis, "On the automatic modularization of software systems using the Bunch tool," *IEEE Trans. Software Engineering*, vol. 32, no. 3, pp. 193-208, Mar. 2006.
[70] J. Grabowski and J. Pempera, "The permutation flow shop problem with blocking. A tabu search approach," *Omega*, vol. 35, no. 3, pp. 302-311, Jun. 2007.
[71] M. Sun, "Solving the uncapacitated facility location problem using tabu search," *Computers & Operations Research*, vol. 33, no. 9, pp. 2563-2589, Sep. 2006.
[72] M.K. Tiwari, S. Kumar, Prakash, and R. Shankar, "Solving part-type selection and operation allocation problems in an FMS: an approach using constraints-based fast simulated annealing algorithm," *IEEE Trans. Systems, Man & Cybernetics, Part A*, vol. 36, no. 6, pp. 1170-1184, Nov. 2006.
[73] G. Attiya and Y. Hamam, "Task allocation for maximizing reliability of distributed systems: a simulated annealing approach," *J. Parallel & Distributed Computing*, vol. 66, no. 10, pp. 1259-1266, Oct. 2006.
[74] M. Moz and M.V. Pato, "A genetic algorithm approach to a nurse rerostering problem," *Computers & Operations Research*, vol. 34, no. 3, pp. 667-691, Mar. 2007.
[75] I.H. Toroslu and Y. Arslanoglu, "Genetic algorithm for the personnel assignment problem with multiple objectives," *Information Sciences*, vol. 177, no. 3, pp. 787-803, Feb. 2007.
[76] A.C. Zecchin, A.R. Simpson, H.R. Maier, M. Leonard, A.J. Roberts, and M.J. Berrisford, "Application of two ant colony optimisation algorithms to water distribution system optimization," *Mathematical & Computer Modelling*, vol. 44, no. 5-6, pp. 451-468, Sep. 2006.
[77] P.Y. Yin and J.Y. Wang, "Ant colony optimization for the nonlinear resource allocation problem," *Applied Mathematics & Computation*, vol. 174, no. 2, pp. 1438-1453, Mar. 2006.
[78] J.S. Heo, K.Y. Lee, and R. Garduno-Ramirez, "Multiobjective control of power plants using particle swarm optimization techniques," *IEEE Trans. Energy Conversion*, vol. 21, no. 2, pp. 552-561, Jun. 2006.
[79] S.H. Jacobson, L.A. McLay, S.N. Hall, D. Henderson, and D.E. Vaughan, "Optimal search strategies using simultaneous generalized hill climbing algorithms," *Mathematical & Computer Modelling*, vol. 43, no. 9-10, pp. 1061-1073, May 2006.
[80] L. You and S. Wood, "Assessing the spatial distribution of crop areas using a cross-entropy method," *Int'l J. Applied Earth Observation & Geoinformation*, vol. 7, no. 4, pp. 310-323, Dec. 2005.
[81] M.A. Arostegui Jr., S.N. Kadipasaoglu, and B.M. Khumawala, "An empirical comparison of tabu search, simulated annealing, and genetic algorithms for facilities location problems," *Int'l J. Production Economics*, vol. 103, no. 2, pp. 742-754, Oct. 2006.
[82] S. Kannan, S.M.R. Slochanal, and N.P. Padhy, "Application and comparison of metaheuristic techniques to generation expansion planning problem," *IEEE Trans. Power Systems*, vol. 20, no. 1, pp. 466-475, Feb. 2005.
[83] E. Elbeltagi, T. Hegazy, and D. Grierson, "Comparison among five evolutionary-based optimization algorithms," *Advanced Engineering Informatics*, vol. 19, no. 1, pp. 43-53, Jan. 2005.
[84] S. Kumar, S.H. Ong, S. Ranganath, and F.T. Chew, "A luminance- and contrast-invariant edge-similarity measure," *IEEE Trans. Pattern Analysis & Machine Intelligence*, vol. 28, no. 12, pp. 2042-2048, Dec. 2006.
[85] J.D. Clayden, M.E. Bastin, and A.J. Storkey, "Improved segmentation reproducibility in group tractography using a quantitative tract similarity measure," *NeuroImage*, vol. 33, no. 2, pp. 482-492, Nov. 2006.
[86] P. Paclik, J. Novovicova, and R.P.W. Duin, "Building road-sign classifiers using a trainable similarity measure," *IEEE Trans. Intelligent Transportation Systems*, vol. 7, no. 3, pp. 309-321, Sep. 2006.
[87] Y. Peng and C.W. Ngo, "Clip-based similarity measure for query-dependent clip retrieval and video summarization," *IEEE Trans. Circuits & Systems for Video Technology*, vol. 16, no. 5, pp. 612-627, May 2006.

[88] F. van der Meer, "The effectiveness of spectral similarity measures for the analysis of hyperspectral imagery," *Int'l J. Applied Earth Observation & Geoinformation*, vol. 8, no. 1, pp. 3-17, Jan. 2006.
[89] M. Popescu, J.M. Keller, and J.A. Mitchell, "Fuzzy measures on the gene ontology for gene product similarity," *IEEE/ACM Trans. Computational Biology & Bioinformatics*, vol. 3, no. 3, pp. 263-274, Jul.-Sep. 2006.
[90] X. Chen, J. Tian, and X. Yang, "A new algorithm for distorted fingerprints matching based on normalized fuzzy similarity measure," *IEEE Trans. Image Processing*, vol. 15, no. 3, pp. 767-776, Mar. 2006.
[91] S. Lee and M.M. Crawford, "Unsupervised multistage image classification using hierarchical clustering with a bayesian similarity measure," *IEEE Trans. Image Processing*, vol. 14, no. 3, pp. 312-320, Mar. 2005.
[92] B. Moghaddam, C. Nastar, and A. Pentland, "A bayesian similarity measure for deformable image matching," *Image & Vision Computing*, vol. 19, no. 5, pp. 235-244, Apr. 2001.
[93] D. Skerl, B. Likar, and F. Pernus, "A protocol for evaluation of similarity measures for rigid registration," *IEEE Trans. Medical Imaging*, vol. 25, no. 6, pp. 779-791, Jun. 2006.
[94] H. Núñez, M. Sànchez-Marrè, U. Cortés, J. Comas, M. Martínez, I. Rodríguez-Roda, and M. Poch, "A comparative study on the use of similarity measures in case-based reasoning to improve the classification of environmental system situations," *Environmental Modelling & Software*, vol. 19, no. 9, pp. 809-819, Sep. 2004.
[95] M. Kirac, G. Ozsoyoglu, and J. Yang., "Annotating proteins by mining protein interaction networks," *Bioinformatics*, vol. 22, no. 14, pp. e260-e270, Jul. 2006.
[96] S. Ray and M. Craven, "Learning statistical models for annotating proteins with function information using biomedical text," *BMC Bioinformatics*, vol. 6, no. 1, pp. rec. S18, May 2005.
[97] J.V. Ponomarenko, P.E. Bourne, and I.N. Shindyalov, "Assigning new GO annotations to protein data bank sequences by combining structure and sequence homology," *Proteins*, vol. 58, no. 4, pp. 855-865, Mar. 2005.
[98] R.M. Othman, S. Deris, R.M. Illias, Z. Zakaria, and S.M. Mohamad, "Automatic clustering of gene ontology by genetic algorithm," *Int'l J. Information Technology*, vol. 3, no. 1, pp. 37-46, Apr. 2006.
[99] R.M. Othman, S. Deris, R.M. Illias, H.T. Alashwal, R. Hassan, and F. Mohamed, "Incorporating semantic similarity measure in genetic algorithm: an approach for searching the gene ontology terms," *Int'l J. Computational Intelligence*, vol. 3, no. 3, pp. 257-266, May 2006.
[100] R.M. Othman, S. Deris, and R.M. Illias, "UTMGO: a tool for searching a group of semantically related gene ontology terms and application to annotation of anonymous protein sequence," *Int'l J. Biomedical Sciences*, vol. 1, no. 2, pp. 111-119, Jul. 2006.
[101] E. Takashima, Y. Murata, N. Shibata, and M. Ito, "Techniques to improve exploration efficiency of parallel self-adaptive genetic algorithms by dispensing with iteration and synchronization," *Systems & Computers in Japan*, vol. 37, no. 14, pp. 25-33, Dec. 2006.
[102] Rahul, D. Chakraborty, and A. Dutta, "Optimization of FRP composites against impact induced failure using island model parallel genetic algorithm," *Composites Science & Technology*, vol. 65, no. 13, pp. 2003-2013, Oct. 2005.
[103] K. Katayama, H. Hirabayashi, and H. Narihisa, "Analysis of crossovers and selections in a coarse-grained parallel genetic algorithm," *Mathematical & Computer Modelling*, vol. 38, no. 11-13, pp. 1275-1282, Dec. 2003.

Razib M. Othman is a doctoral candidate at the Faculty of Computer Science and Information System, the Universiti Teknologi Malaysia. He received the BSc and MSc degrees in Computer Science both from the Universiti Teknologi Malaysia, in 1999 and 2003 respectively. Currently, he is working for his PhD in Computational Biology. He also has interests in artificial intelligence, software agent, parallel computing, and web semantics. In March 2005, he was awarded the Young Researcher award by the Malaysian Association of Research Scientists (MARS). Two of his inventions, software products named *2D Engineering Drawing Extractor* and *2D Design Structure Recognizer*, have won 5 awards at the 21st Invention and New Product Exposition held in Pittsburgh, USA including the Best Invention of the Pacific Rim, and a gold medal award at the 34th International Exhibition of Inventions of New Techniques and Products held in Geneva, Switzerland.

Safaai Deris is a Professor of Artificial Intelligence and Software Engineering at the Faculty of Computer Science and Information Systems, Deputy Dean at the School of Graduate Studies, and Director of Laboratory of Artificial Intelligence and Bioinformatics at the Universiti Teknologi Malaysia. He received the MEng degree in Industrial Engineering, and the DEng degree in Computer and System Sciences, both from the Osaka Prefecture University, Japan, in 1989 and 1997 respectively. His recent academic interests include the application and development of intelligent techniques in planning, scheduling, and bioinformatics.

Rosli M. Illias is an Associate Professor at the Faculty of Chemical and Natural Resources Engineering at the Universiti Teknologi Malaysia. He received the PhD degree in Molecular Biology from the Edinburgh University, UK in 1997, and the BSc degree in Microbiology from the Universiti Kebangsaan Malaysia in 1992. His research interests are in the areas of microbial technology, molecular enzymology, and molecular genetics.