A Modular Database Architecture Enabled to Comparative Sequence Analysis

Paola Bonfante[1], Francesca Cordero[2,3], Stefano Ghignone[1], Dino Ienco[2], Luisa Lanfranco[1], Giorgio Leonardi[4], Rosa Meo[2], Stefania Montani[4], Luca Roversi[2], and Alessia Visconti[2]

[1] Dipartimento di Biologia Vegetale, Università di Torino, Italy
[2] Dipartimento di Informatica, Università di Torino, Italy
[3] Dipartimento di Scienze Cliniche e Biologiche, Università di Torino, Italy
[4] Dipartimento di Informatica, Università del Piemonte Orientale, Italy
{fcordero,ienco,meo,roversi,visconti}@di.unito.it,
{giorgio.leonardi}@mfn.unipmn.it,
{stefania.montani}@unipmn.it,
{pbonfant,sghignon,llanfran}@unito.it

Abstract. The beginning of post-genomic era is characterized by a rising numbers of public collected genomes. The evolutionary relationship among these genomes may be caught by means of the comparative analysis of sequences, in order to identify both homologous and non-coding functional elements. In this paper we report on the on-going BIOBITS project. It is focused on studies concerning the bacterial endosymbionts, since they offer an excellent model to investigate important biological events, such as organelle evolution, genome reduction, and transfer of genetic information among host lineages. The BIOBITS goal is two-side: on the one hand, it pursues a logical data representation of genomic and proteomic components. On the other hand, it aims at the development of software modules allowing the user to retrieve and analyze data in a flexible way.

1 Introduction

Genomics and post-genomics studies which have bloomed in the last decade are offering new tools for applied biotechnological research in several fields from medical, pharmaceutical to industrial and environmental. Sequencing of the human genome has generated a great deal of interest in the diagnosis and treatment of diseases using genomic medicines. Structural genomics approaches covering topologically similar proteins or gene families are great assets for progress in the development of novel therapeutics. In addition the genomic analysis of microbial communities in a culture-independent manner (metagenomics) has also given the opportunity to probe and exploit the enormous resource represented by still underscribed microbial diversity.

A. Hameurlain et al. (Eds.): TLDKS IV, LNCS 6990, pp. 124–147, 2011.

This paper is an extension of a work already published [12]. It describes the on-going project BIOBITS[1] that aims at performing an extensive comparative genomic studies in order to answer fundamental questions concerning the biology, ecology and evolutionary history. The specific goal of BIOBITS is to get insights on the tri-partite system, constituted by (i) a bacterial endosymbiont of an arbuscular mycorrhizal (AM) fungus, (ii) AM fungi living in plant roots, and (iii) plant roots.

Bacterial endosymbionts are widespread in the animal kingdom, where they offer excellent models for investigating important biological events such as organelle evolution, genome reduction, and transfer of genetic information among host lineages [30]. By contrast, examples of endobacteria living in fungi are limited [26] and those best investigated live in the cytoplasm of AM fungi [9]. AM fungi are themselves obligate symbionts since, to complete their life cycle, they must enter in association with the root of land plants.

AM species belonging to the Gigasporaceae family harbour an homogeneus population of endobacteria which have been recently grouped into a new taxon named *Candidatus* Glomeribacter gigasporarum [7]. The AM fungus and its endobacterium Ca. Glomeribacter gigasporarum are currently used as a model system to investigate endobacteria-AM fungi interactions.

The project takes advantages by the employment of a massive large-scale analysis and genomic comparison study of phylogenetically related free-living bacteria. Moreover, the comparison with genomes of other endosymbionts species will provide insights about the reason of the strict endosymbiotic life-style of this bacterium.

Another aspect taken into account is the analysis of metabolic pathways. A strong reason of interest in this project is based on the assumption that the symbiotic consortia may lead to the discovery of molecules of interest for the development of novel therapies and other applications in biotech.

In this paper we report specifically on a step of BIOBITS whose goal, roughly, is the development of a modular database which allows to import, to store, and to analyze massive genomic data. Later in BIOBITS we will extensively develop a computational genomic comparison focused on the above bacterium and fungi genomes. BIOBITS deploys a data warehouse that stores in a multi-dimensional model the interesting components of the project. Such a component should have the following characteristics: i) being able to store genomic data from multiple organisms, possibly taken from different public database sources; ii) annotating the genomic data making use of the alignment between the given sequences and the genomic sequences of other similar organisms; iii) annotating the genomic sequences and the protein transcript products by the full use of ontologies developed by the biology and bioinformatics communities; iv) comparing and visually presenting the results of the genomic alignment; iv) being able to cluster genomic or proteomic data coming from different organisms. The aim is at finding easily

[1] BIOBITS is a project funded by Regione Piemonte under the Converging Technologies Call. BIOBITS involves Università di Torino, Università del Piemonte Orientale, CNR and the companies ISAGRO Ricerca s.r.l., GEOL Sas, Etica s.r.l.

increasing levels of similarity and induce on one side the steps of the phylogenetic evolution and on the other side investigate on the metabolic pathways.

As a matter of fact, we wish to take advantage of the possibilities offered by computer science technology and its methodologies to analyse the genomic data the project will produce. The analysis of genomic data requires computational tools that allow to "navigate" flexibly data from arbitrary (at least in principle) user defined perspectives and under different degrees of approximation.

In this paper we describe the BIOBITS system architecture in terms of BIOBITS Data Mart and BIOBITS modules. With respect to the previous publication [12], we report a detailed description of two modules, namely *Case Base Reasoning* and *Co-clustering* modules, that have been developed to perform a comparative genomic analysis. Moreover, we show the results obtained in a case study by the use of the system. The case study shows the utility and flexibility of an integrated system whose modules allow to retrieve and analyze different portions of data, at the granularity level that is needed by the user. This flexibility eliminates the necessity of perform any pre-processing to the data in order to adapt it to the analysis algorithm and to the user's goal.

In the presented case study we extracted a set of biological sequences belonging to the organism under investigation by following the BIOBITS Data Mart star schema. BIOBITS project focus on the identification of the evolutionary relationships among species more similar to *Ca. G. gigasporarum*. Using the Case Base Reasoning module, we retrieved sequences that are similar to the given organism. Retrieval is performed according to the suitable abstraction level over the data given by a taxonomy of granularities. Finally, to the resulting sequences we applied the Co-clustering module and we were able to identify protein domains common among the sequences.

2 Related Works

There is a wide variety of approaches in designing tools to analyze biological data. Experience suggests that the best way to data analysis is to set up a database. An 'historical' example is ACeDB (*A C. elegans Database* [1]), one of the first hierarchical, rather than relational, model organism databases. Another example is ArkDB [21], a schema that was created to serve the needs for the subset of the model organism community interested in agriculturally important animals. ArkDB has been successfully used across different species by different communities, but is rarely used outside the agricultural community.

On "top" of databases a great variety of applications is available, from those ones for the annotation community to molecular pathway visualization, or from the work-flow management to the comparative genome visualization.

Currently, there is a rich community and many available software tools built around MAGE [27] and GMOD [33]. GMOD stands for Generic Model Organism Database project, which brought to the development of a whole collection of software tools for creating and managing genome-scale biological databases, in the forthcoming description. In the BIOBITS project GMOD and its database Chado have been selected as the data elaboration and management center.

2.1 GMOD and Chado Database

The BIOBITS software architecture is built upon a layer provided by GMOD system. We report here the main motivations that lead to this choice.

The design and implementation of database applications is time consuming and labor-intensive. When database applications are constructed to work with a particular schema, changes to the database schema require in turn changes to the software. Unfortunately, these changes are frequent in real projects due to changes in requirements. In particular they are frequent in bioinformatics. Most critical are the changes in the nature of the underlying data, which follow the current understanding of the natural world. Additional requirements are placed by the rapid technological changes in experimental methods and materials. Finally, the wide variety of biological properties in the organisms species always has made difficult to create a unique model schema valid for all the species.

All the above outlined motivations led to the design of Chado database model which is a generic and extensible model, whose software is available under an open source delivery policy. Chado schema can be employed as the core schema of any model organism data repository. This common schema increases interoperability between software modules that operate on it.

Chado data population is driven by *ontologies*, i. e. *controlled vocabularies*. Ontologies give a typing to the entities with the result of partitioning the whole schema into *subschemas*, called *modules*. Each *module* encapsulates a different biological domain and uses an appropriate ontology. An ontology characterizes the different types of entities that exist in a world under consideration by means of primitive relations. These primitives are easy to understand and to use, they are expressive and consistent, and they allow the reasoning about the concepts under representation. Typical examples of ontological relations are: (i) *is_a* which expresses when a class of entities is a subclass of another class, and (ii) *part_of* which expresses when a component constitutes a composite. Many other relation types are discussed in [15].

Concerning the schema of Chado it is worth remarking *feature* and *sequence* entities. *feature* allows both data and meta data; it can be populated by instances each determining the type of every other instance in the schema, in accordance with the ontology SO [15]. *sequence* contains biological sequence features, that include genetically encoded entities like genes, their products, exons, regulatory regions, etc... *feature* and *sequence* are further described by properties.

3 BIOBITS System Architecture

Here we deepen the description of the system which is designed to manage all the information and all the in-silico activities in the context of the project BIOBITS. This system is implemented through a modular architecture, described in detail in Section 3.2. The system architecture permits (1) to store and access locally all the information regarding the organisms to be studied, and (2) to provide algorithms and user interfaces to support the researchers' activities like: (i) searching and retrieving genomes, (ii) comparing and aligning with a genome

of reference, (iii) investigating syntenies, and (iv) locally storing potentially new annotations.

The system architecture has been engineered exploiting the standard modules and interfaces offered by the GMOD project [33], and completed with custom modules to provide new functionalities. The main module of the system contains the database which provides all the data needed to perform the in-silico activities related to the project.

Thanks to the adoption of Chado database schema, on the one hand, we take advantage of its support in controlled vocabularies and ontologies. On the other hand, Chado is the standard database for most of the GMOD modules; therefore we can reuse these modules to support the main activities of the project and extend the system incrementally as the researchers' needs evolve. An example, is the possibility to use BioMart Chado's module which helps the user to identify the relevant dimensions of the problem, their hierarchies and to transform and import input data in the data warehouse conforming them in a typical star schema.

3.1 Star Schema in BIOBITS Data Mart

Essential in the data warehouse is the logical star schema of the stored data. The star schema defines the dimensions of the problem. Often, each dimension of the star schema can be viewed at different abstraction levels. The levels are organized in a hierarchy. Finally, the central entity in the star schema collects the main facts or events of interest. In the case of the BIOBITS project, there are two star schemas.

1. The star built around the *genome composition* facts. It represents the composition of each genome in terms of genes and chromosomes and with reference to the belonging organism.
2. The star schema around *protein* facts. It describes the proteins in terms of PROSITE domains and with respect to the dimensions of phylogenetic classification and metabolic pathways.

The genes and proteins facts are linked by the relationship representing the encoding.

For most of the dimensions, such as genes and phylogenetic classification, the scientific literature already has provided ontologies (e.g., Gene Ontology, GO) and controlled vocabularies (Clusters of Orthologous Groups, COG) that are available in public domain databases and are imported in the system. Another example of available hierarchy on the genes and proteins are the family organizations.

In the following we describe the BIOBITS Data Mart schema (shown in Figure 1) in detail.

Genome Composition. It includes all the relevant information about a genome fragment. Considering a fragment view of the genome, genome composition includes all the known fragments composing a genome: it reports the precise boundaries of the fragments (which depend on the user experience and

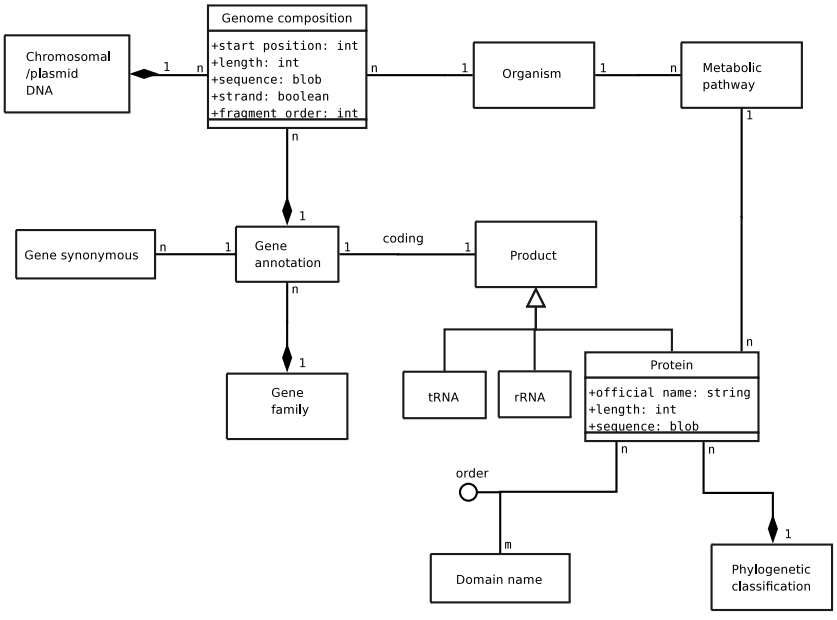

Fig. 1. Star schema of BIOBITS Data Mart

discoveries), the start position and the fragment order with respect to the genome, its nucleotide sequence and strand.

Chromosome/Plasmid DNA. It specifies the localization of the fragment expressed by the number or the name of the corresponding chromosome/plasmid location. Indeed, the genome could be inserted either in a chromosome sequence or in a plasmid sequence.

Organism. It specifies both endosymbiotic and ectosymbiotic bacteria. An organism is identified by the specified identifier, includes the organism scientific name and its classifications in the taxonomy database.

Gene Annotation. It consists in a short report of gene-specific information (identifier and name), comprehensive of a brief description of gene products using both the information reported in Gene Ontology, and the main references stored in `Pubmed`.

Gene Synonymous. It contains all the synonymous names associated to each gene. Genes and proteins are often associated to multiple names; additional names are included as new functional or structural information are discovered. Since authors often alternate between synonyms, computational analysis benefits from collecting synonymous names.

Gene Family. Following the gene classification into families, consistent to the genes biochemical similarity, it reports the family identifiers.

Product. It is a class of the products that genes codify. Products are categorized into in three classes: transfer RNA (tRNA), ribosomal RNA (rRNA) and proteins. Moreover, it reports a pseudogene indication if the gene has lost its coding ability.

tRNA. Transfer RNA is a small RNA molecule that transfers a specific active amino acid to a growing polypeptide chain.

rRNA. Ribosomal RNA is the central component of the ribosome. The ribosome is a complex of ribosomal RNA and ribonucleoproteins.

Metabolic Pathways. It represents pathways which are composed by a set of biochemical reactions. Each pathway represents the knowledge on the molecular interactions and reactions network.

Protein. It refers to protein-specific information (protein identifier and name). A protein is a set of organic compounds (polypeptides) obtained by transcription and translation of a DNA sequence.

Phylogenetic Classification. It consists of Cluster of Ortologous Groups (COG) of protein sequences encoded in a complete genome.

Domain Name. It reports the domains extracted from PROSITE database [22], characterizing the protein sequence. PROSITE consists of documentation entries describing protein domains, families and functional sites.
The relationship among proteins and domains is characterized by the attribute *order* describing how the domains that compose a specific protein are sorted.

3.2 System Architecture

Figure 2 summarizes the main architecture of the BIOBITS system. In the following we focus on objectives and features of the BIOBITS system.

Local and global access to data. The instance of Chado we want to set up will contain both data on genome we shall explicitly produce as part of the project BIOBITS and data retrieved from the biological databases accessible through the Internet. The `Import modules` in Figure 2 will accomplish such a requirements. Concerning the retrieval from Internet, *RRE - Queries* is a GUI wizard, built on the basis of a previously published tool [24], able to query different biological databases like for example `GenBank` [19] and able to convert the results of the queries into standard formats. Alternatively, we can convert the format of data retrieved from Internet thanks to the scripts available as part of the GMOD project. A remarkable example are those scripts that convert `GenBank` genes annotations into the Generic Feature Format (`GFF`), adopted as a standard in the GMOD project. Of course, once data have been retrieved, *Import Modules* update Chado, either on-demand, or automatically, possibly on a regular basis.

An On Line Architecture Mining architecture. One of the advantages of a data warehouse is the ready availability of clean, integrated and consolidated data represented by a multiplicity of dimensions. Once that data are stored in the data warehouse, elementary statistics can be computed on the available facts

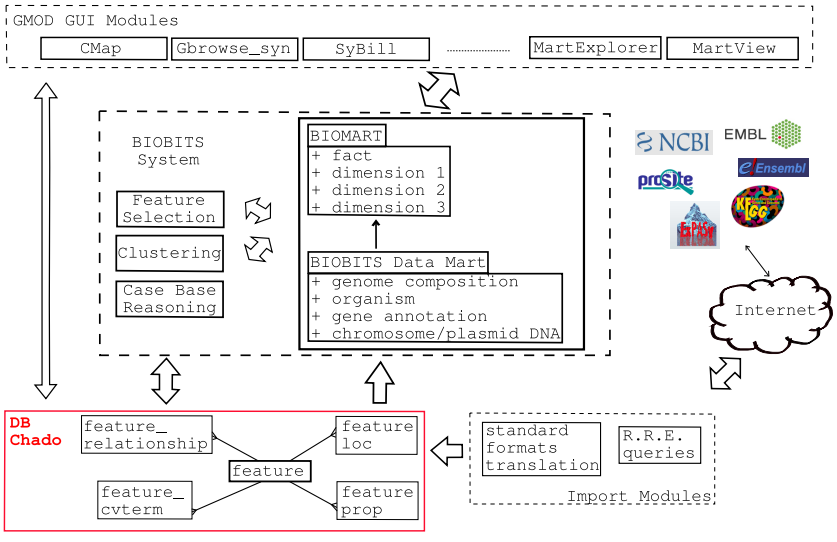

Fig. 2. The architecture of BIOBITS system

and aggregation of measures and frequencies of facts can be immediately computed. The results can be browsed and compared by OLAP primitives and tools. Finally, on these statistics the power of data mining algorithms can be further exploited. This is the On Line Architecture Mining (OLAM) view of a software architecture [20]. OLAM is composed by a suite of data mining algorithms that receive from the client a query for a knowledge discovery task. The request can be answered by the predictive and semi-automatic capabilities of data mining algorithms. In turn, these ones work on the results of an underlying OLAP server that receives the input data from the underlying data warehouse.

For the transformation of the data stored in Chado into the star schema of Figure 1 we exploit BioMart [8], which is a software package available inside GMOD.

Services on Chado and the Star Schema. In Figure 2, associated to both the Chado instance and to the BIOBITS Data Mart we plan to offer two types of services. The first type is implemented on the basis of existing modules of GMOD. Figure 2 highlights them in the uppermost dashed box, named GMOD `GUI Modules`. The second type of services are internal to the real BIOBITS system: they are shown in Figure 2 inside the central dashed box, named BIOBITS system. Now, we discuss the latter components in detail, putting much emphasis on the features of the software modules that we specifically develop in support to the realization of the goals of the project.

GMOD Graphical User Interface Modules. These modules exploit the available GMOD modules using Chado database to provide the researchers with the tools for comparative genomics needed by the BIOBITS project. GUI modules have

also a graphical user interface and allow the user to interact with the system. In particular,

- `CMap` allows users to explore comparisons of genetic and physical maps. The package also includes tools for maintaining map data;
- `GBrowse` is a genome viewer, and also permits the manipulation and the display of annotations on genomes;
- `GBrowse_syn` is a GBrowse-based synteny browser designed to display multiple genomes, with a central reference species compared to two or more additional species;
- `Sybil` is a system for comparative genomics visualizations;
- `MartExplorer` and `MartView` are two user interfaces allowing the user to explore and visualize the stored experimental results and the database content.

BIOBITS system specific modules. The goal of these modules is to allow data analysis under two perspectives that should complement each other and serve for validation.

The first perspective is the one offered by the *Case Base Reasoning* module. It supports efficient retrieval strategies in the context of the search for genomic similarity and syntenies, directly operating on our implementation of the star schema inside BioMart.

The other perspective will exploit tools from Data Mining. We shall use them to perform advanced elaboration on the genomic data. Among the data mining modules we foresee modules for classification, for feature selection and clustering. The latter will be discussed in more detail in this paper, since it has been the first to be integrated into the BIOBITS system. Indeed, one of the main goal of the whole BIOBITS project is to provide the results of fragment alignment tools. Since clustering provides a specifically useful service for the exploration and elaboration of the similarities among genes and proteins, its results could provide to the syntheny tools additional information that would enhance the fragment elaboration.

As a concluding remark, the plan is to develop BIOBITS system specific modules as web-based GUI in order to gain user-friendliness and a good degree of interoperability, similar to current GMOD modules that are able to connect to other modules by standard interfaces.

Of course we shall adhere to the open source philosophy. So, any BIOBITS system specific module will be available as part of the whole project GMOD.

4 Software Modules to Support Researchers' Activities

The main contribution of the BIOBITS project is the development of two GMOD modules to analyse the knowledge stored in the data warehouse. The following section describes the details of these new modules based on Case Based Reasoning and clustering.

4.1 Case-Based Reasoning

Within the BIOBITS architecture, we worked at the design and implementation of an *intelligent retrieval* module, which implements the *retrieval* step of the Case-Based Reasoning (CBR) [2] cycle. CBR is a reasoning paradigm that exploits the knowledge collected on previously experienced situations, known as *cases*. The CBR cycle operates by (1) *retrieving* past cases that are similar to the current one and by (2) *reusing* past successful solutions; (3) if necessary, past solutions are properly *adapted* to the new context in which they have to be used; finally (4) the current case can be *retained* and put into the system knowledge base, called the *case base*. It is worth noting that *purely retrieval* systems, leaving to the user the completion of the reasoning cycle (steps 2 to 4), are very valuable decision support tools [38], especially when automated adaptation strategies can hardly be identified, as in biology and medicine [28]. This is exactly the strategy we are following in the current approach.

Our retrieval module is meant to support comparative genomics studies that represent a key instrument to: (1) discover or validate phylogenetic relationships, (2) give insights on genome evolution, and (3) infer metabolic functions of a particular organism. In the module, cases are genomes, each one taken from a different organism, and properly aligned with the same reference organism. Indeed, the alignment task is a prerequisite in our library. For this reason we start describing the selected sequences alignment strategy, then we detail our module deep down into the cases representation and retrieval.

Sequence Alignment. To deal with the alignment task we rely on BLAST [3]. BLAST is a state-of-the-art local alignment algorithm, specifically designed for bioinformatics applications. It takes as an input a sequence of nucleotides and properly aligns it to a database of strings belonging to (different) organisms of interest.

From a typical BLAST output (Figure 3) one can extract basic information (percentage of the sequence that shows identity and length of the sequence alignment) that can be easily plotted as represented in Figure 4.

Case Representation. From an application viewpoint, it makes sense to convert the *quantitative* similarity values in Figure 4 to a set of *qualitative* levels (e.g. low, medium, high similarity). This provides a "higher level" view of the information, able to abstract from unnecessary details. To perform the conversion, we exploit a semantic-based abstraction process, similar to the Temporal Abstractions (TA) techniques, described in [40,5]. Indeed, in our domain, we consider as the independent variable the symbol position in the aligned strings, instead of the time. As in TA we move from a *point-based* to an *interval-based* representation of data, where the input points are the symbol positions, and the output intervals (*episodes*) aggregate adjacent points sharing a common behavior, persistent over the sequence. In particular, we rely on *state* abstractions [5], to extract episodes associated with qualitative levels of similarity between the two aligned strings, where the mapping between qualitative abstractions

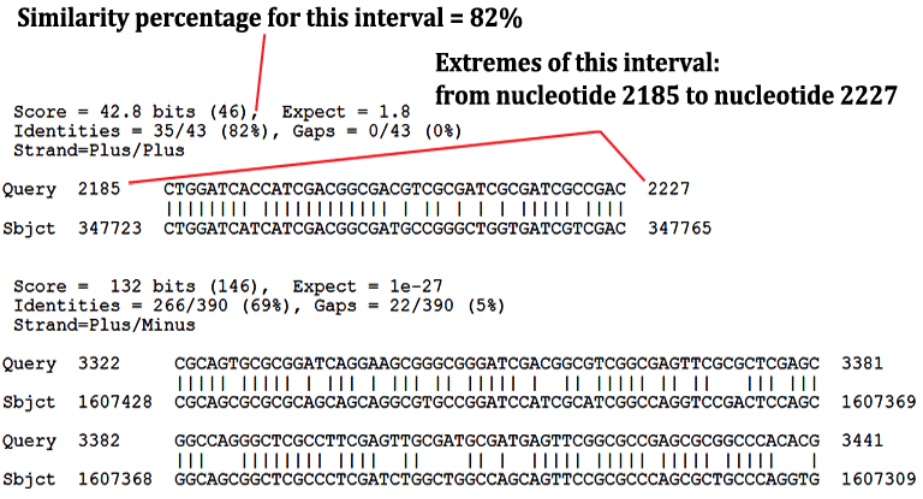

Fig. 3. BLAST sequence alignment

and quantitative values of the similarity has to be parametrized on the basis of domain semantic knowledge. Semantic knowledge can also support a further refinement of the state abstraction symbols, according to a taxonomy like the one described in Figure 5. Obviously, the taxonomy can be properly modified depending on specific domain needs.

Moreover, our tool allows the representation of the available sequences at any level of detail, according to a taxonomy of granularities, like the one depicted in Figure 6. This granularity change makes sense from a biological point of view: consider e.g. that a region may be conserved among relative organisms, while a specific gene within the region may not. Thus, a high similarity at the region level might be difficultly identified at the level of single genes (as it will be shown in the example discussed in Subsection 5.1).

Notice that the taxonomy of the granularities definition is strongly influenced by domain semantics. For instance, the number of nucleotides which composes a gene depends on the specific organism, and on the specific gene. Domain knowledge also strongly influences the conversion of a string of symbols from a given granularity to a different one, as required for flexible retrieval.

To summarize, *case representation* is obtained as follows. First, an optimal alignment of two nucleotide strings is calculated by BLAST. In particular, for each subsequence of nucleotides, a percentage of similarity with the aligned nucleotide in the paired string is provided. Abstractions on such quantitative levels are then calculated, and allow to convert these values into qualitative ones, expressed as strings of symbols. Abstractions are calculated at the ground level in the symbol taxonomy (and operate also at the ground level in the granularity taxonomy, since they work on nucleotides, see Figure 6). The resulting string of symbols is then stored in the case library as a *case*. Despite the fact that cases are stored as abstractions at the ground level, they could be easily converted at

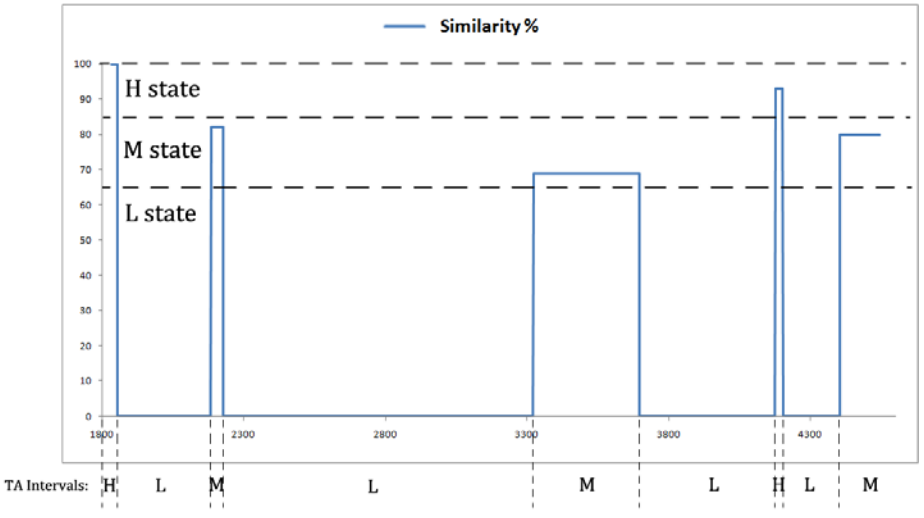

Fig. 4. A graphical visualization of sequence alignment

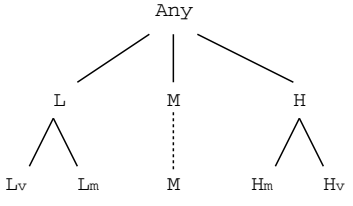

Fig. 5. An example taxonomy of state abstraction symbols; for instance, the high (H) symbol specializes into very high (H_v) and moderately high (H_m)

coarser levels in both dimensions (i.e. the dimension of the taxonomy of symbols, and the one of granularities). Such conversion is the means by which we support flexible case retrieval and will be described below.

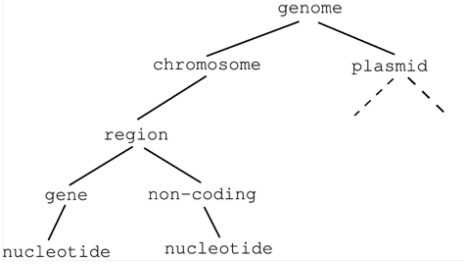

Fig. 6. A taxonomy of sequence granularities

Case Retrieval (query answering). Taking advantage from the *multi-level abstraction* representation introduced above, we support *flexible retrieval*.

In particular, we allow users to express their queries for case retrieval at any level of detail, both in the dimension of data descriptions (i.e. at any level in the taxonomy of symbols) and in the dimension of the granularity.

Obviously, since cases are stored at the ground level in both dimensions, in order to identify the cases that match a specific query, the analyst must provide a function for scaling up (*up* henceforth) two or more symbols expressed at a specific granularity level to a single symbol expressed at a coarser one. Moreover, a proper distance function must be defined.

The data structures described above, as well as the *up* and the distance functions, have to be detailed on the basis of the semantics of the specific application domain. However, we have identified a set of general "consistency" constraints, that any meaningful choice must satisfy, in order to avoid ambiguous or meaningless situations. For instance, we enforce the fact that distance monotonically increases with ordering in the symbol domain.

Moreover, distance "preserves" ordering also in the case in which *is_a* relationships between symbols are involved. For example, the distance between L (low) and M (medium) is smaller than the distance between L (low) and H_v (very high). The exhaustive presentation of such constraints is outside the scope of this paper, but can be found in [29].

In order to increase efficiency, our framework also takes advantage of *multi-dimensional orthogonal index structures*, which allow for early pruning and focusing in query answering. Indexes are built on the basis of the data structures previously described. The root node of each index is a string of symbols, defined at the highest level in the symbol taxonomy, i.e. the children of "Any", as shown in Figure 5, and in the granularity taxonomy. A –possibly incomplete, index stems from each root, describing refinements along the granularity and/or the symbol dimension. An example multi-dimensional index, rooted in the H symbol, is represented in Figure 7. Note that, in the figure, granularity has been chosen as the *leading dimension*, i.e. the root symbol is first specialized in the granularity dimension. From each node of the resulting index, the sequence of the symbols of the node itself is then orthogonally specialized in the *secondary* (i.e. the symbol) *dimension*, while keeping granularity fixed. However, the opposite choice for instantiating the leading and the secondary dimensions would also be possible.

Each node in each index structure is itself an index, and can be defined as a *generalized case*, in the sense that it summarizes (i.e. it indexes) a set of cases. This means that the same case is typically indexed by different nodes in one index (and in the other available indexes). This supports flexible querying, since, depending on the level at which the query is issued, according to the two taxonomies, one of the nodes can be more suited to provide a quick answer.

To answer a query, to enter the more proper index structure, we first progressively generalize the query itself in the secondary dimension (i.e. the symbol taxonomy in the example), while keeping the leading dimension (i.e. granularity

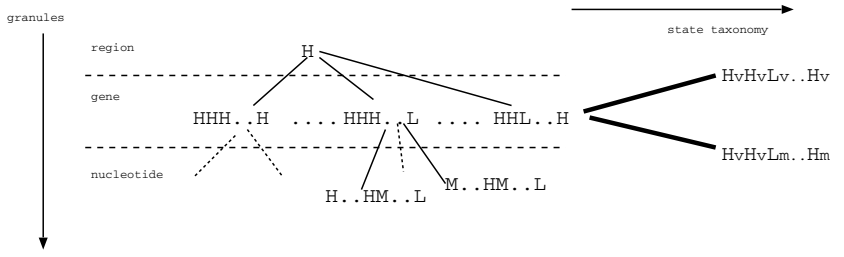

Fig. 7. An example of a multi-dimensional orthogonal index

in the example) fixed. Then, we generalize the query in the other dimension as well. Following the generalization steps backwards, we can enter the index from its root, and descend along it, until we reach the node which fits the leading dimension level of the original query. If an orthogonal index stems from this node, we can descend along it, always following the query generalization steps backwards. We stop when we reach the same detail level in the secondary dimension as in the original query. If the query detail level is not represented in the index, because the index is not complete, we stop at the most detailed possible level. We then return all the cases indexed by the selected node.

It is worth noting that indexes may be incomplete with respect to the taxonomies. Index refinement can be automatically triggered by the storage of new cases in the case base, and by the types of queries which have been issued so far. In particular, if queries have often involved, e.g. a symbol taxonomy level which is not yet represented in the index(es), the corresponding level can be created. A proper frequency threshold for counting the queries has to be set to this end. This policy allows to augment the indexes discriminating power only when it is needed, while keeping the memory occupancy of the index structures as limited as possible.

As a last remark, a number of tools to support comparative genomics studies are already available. For example, the VISTA tool (http://genome.lbl.gov/vista/index.shtml) allows the visualization of pre-computed pairwise and multiple alignments of whole genome assemblies. Our tool, beside alignments visualization, also allows to mine genomes at multiple levels: customized searches can be performed, to retrieve genomes and/or genomic segments matching specific features as described by the query at the desired granularity. Furthermore, thanks to this tool, queries can be performed efficiently and potentially on very large databases. The novelties introduced are exemplified in section 5, with the addition of a performance study.

4.2 Clustering Modules

In this paper we do not go in detail in describing all the predictive and exploratory capabilities offered by data mining algorithms.

The aim of this section is to depict a portrait built on a single example: *clustering*. It offers the possibility to show the benefits in terms of interoperability,

extendability and flexibility offered by a modular system built upon a data warehouse in which a multi-dimensional representation of a ground set of facts is stored. On these data, whenever it is needed, a query can be issued by the user in order to retrieve from the data warehouse the values of the interesting subset of dimensions. On this initial set of values multi-level reasoning is possible exploiting the relationships between facts in the knowledge network.

One of the classical aims of clustering is to provide a description of the data by means of an abstraction process. In many applications, the end-user is used to study natural phenomena by the relative proximity relationships existing among the analyzed objects. For instance, he/she compares organisms by means of the relative similarity in terms of the common features with respect to a same referential example. Many Hierarchical Clustering (HC) algorithms have the advantage that are able to produce a dendrogram which stores the history of the merge operations (or split) between clusters. Moreover, the dendrogram produced by a hierarchical clustering algorithm constitutes a useful, immediate and semantic-rich conceptual organization of the object space. As a result HC algorithms produce a hierarchy of clusters and the relative position of clusters in this hierarchy is meaningful because it implicitly tells the user about the relative similarity between the cluster elements. HC approaches help the experts to explore and understand a new problem domain. As regards the exploitation of object distances, clustering algorithms offer immediate and valuable tools to the end-user for the biological analysis.

Co-clustering. A kind of clustering algorithm particularly useful in biological domains is *co-clustering* [14] whose solution provides contemporaneously a clustering of the objects and a clustering of the attributes. Further, often co-clustering algorithms exploit similarity measures on the clusters in the other dimension of the problem: that is, clusters of objects are evaluated by means of the clusters on the features and vice versa. They simultaneously produce a hierarchical organization in two of the problem dimensions: the objects and the features that describe the objects themselves. In many applications both hierarchies are extremely useful and are searched for.

In a more formalized view, a co-clustering algorithm is an unsupervised data mining method that computes a *bi-partition* of a dataset $X \in \mathbb{R}^{n \times m}$. A bi-partition of a dataset is a triple $\langle R, C, \psi \rangle$, where R is a partition of rows (object instances) into $|R|$ subsets, C is a partition of columns (object attributes) into $|C|$ subsets, and ψ is a relation that associate elements of R to elements of C.

An extension of the algorithm based on co-clustering has been obtained by the introduction of *constraints*. Constraints are very effective in many applications, including gene expression analysis [34] and sequence analysis [13], since the user can express which type of biological knowledge leads to the association among the clusters of genes (the *objects*) and the clusters of biological conditions (the *attributes*).

The goal of the constrained co-clustering algorithm is to find a bi-partition such that a given objective function is optimized and a set of user-defined constraints are satisfied. Two kinds of constraints, i.e. *must-link* and *cannot-link*,

should be exploited. A *must-link* constraint specifies that two rows (respectively, columns) of X must belong to the same cluster. Conversely, a *cannot-link* constraint specifies that two rows (respectively, columns) of X cannot belong to the same cluster.

In general, the satisfaction of constraints may decrease the theoretical optimum of the objective function. Notice also that the satisfaction of a conjunction of constraints is not always feasible. A constrained co-clustering algorithm works as follows. During each iteration, it associates each row to the nearest row cluster which does not violate any cannot-link constraint. If a row is involved in a must-link constraint the algorithm associates the whole set of rows involved in this constraint to the selected row cluster. Furthermore, it controls that any cannot-link constraint is not violated. This process is iterated until the function reaches a desired value, i.e. its decrease is smaller than a user defined threshold τ. The same process is simultaneously performed over the columns of the matrix.

5 Case Study

The recent efforts of several sequencing projects to explore the genomes of organisms from various lineages have provided great resources for comparative genomics. Since the beginning of the postgenomic era, investigators faced how to manage the rising number of public collections of genomes in novel ways [16]. Other than the public databases where sequences are deposited, more specific data warehouses have been developed [23] were the incorporated data types include annotation of (both protein and non-protein coding) genes, cross references to external resources, and high throughput experimental data (e. g. data from large scale studies of gene expression and polymorphism visualised in their genomic context). Additionally, on such platforms, extensive comparative analysis could be performed, both within defined clades and across the wider taxonomy. Furthermore, sequence alignments and gene trees resulting from the comparative analysis can be accessed. Computational challenges in the field of comparative analyses have been overcome [39]. The developed tools have helped in elucidating the genomic structures of a multiple levels of prokaryotes [6], leading to a much improved understanding of why a bacterial genome is organized in the way it is.

A number of comparative analyses closer to our field of investigation have already shade lights on the characterization of genomes of host-associated and free-living bacteria [41,4,32,11,10]. Novel computational approaches on large scale datasets provide a new viewpoint for whole genome analysis and bacterial characterization. For example, the self-attraction clustering approach allowed classification of Proteobacteria, Bacilli, and other species belonging to Firmicutes [35], whereas the research of protein [18] or genomic [37] signatures have been useful to elucidate the evolutionary relationships among the Gammaproteobacteria and to provide new insights into the evolution of symbiotic diversity, microbial metabolism and host-microbe interactions in sponges.

One major focus of comparative sequence analysis is the search for syntenies. The term synteny is used to mean a set of genes that share the same relative

ordering on the genome of different species. In BIOBITS project we are interested on a synteny between several species in order to recognize which are the species more similar to *Ca. G. gigasporarum*. The evolutionary relationships of these genomes may allow the identification of homologous genes and non-coding functional elements, such as regulatory elements and protein domains.

To reach this purpose we exploit the BIOBITS system architecture (shown in Section 3) and the Chado modules described in this paper (see Section 4). To show the reliability of our approach we perform a sequence analysis on a well-known bacterial genus.

5.1 Querying for Synthenies on the Region DCW

Following the Data Mart star schema reported in Figure 1, the data related to a bacterium belonging to the genus of *Burkholderia* (i.e. *Burkholderia xenovorans*) has been extracted. In details, four tables of the Chado database (i.e. *Gene family*, *Gene annotation*, *Genome composition*, and *Organism*) are exploited to extract genes belonging to a specific region called Division Cell Wall (DCW). This region is involved in the synthesis of peptidoglycan precursors and cell division. *DCW cluster* is composed of 14 genes: FtsA, FtsI, FtsL, FtsQ, FtsW, FtsZ, mraW, mraY, mraZ, murC, murD, murE, murF, murG. The prominent feature of the *DCW cluster* is that it is conserved with an high (H) similarity in many bacterial genomes over a broad taxonomic range. Specifically, notwithstanding some bacteria belonging *Burkholderia xenovorans* simply miss one of the 14 genes, all of them maintain a high similarity at the DCW region level with their relatives.

Suppose that a user, interested in comparing bacteria on the basis of the DCW cluster content, asks the flexible retrieval system (see section 4) the following query:
$H_v H_v L_v H_v H_v H_v H_v H_v H_v H_v H_v H_v H_v H_v$
looking for the specific bacteria missing the third gene, but very similar to the reference one as regards the other genes. The flexible retrieval system will first generalize the query in the symbol taxonomy dimension (see Figure 5), providing the string: $HHLHHHHHHHHHHH$
and then in the granularity dimension, providing the query H at the region level. Quite naturally, we define the *up* function as:
$up(HHLHHHHHHHHHHH) = H$.

This allows to enter the index in Figure 7 from its root. Then, following the generalization step backwards, a node identical to the query can be found, and the ground cases indexed by it can be retrieved.

Interactive and *progressive* query relaxation (or refinement) are supported as well in our framework. In this situation the distance between the original query and the cases indexed by the other children of the node can be calculated by any distance function which satisfies the constraints illustrated in [29], and quickly described before. Query relaxation or refinement can be repeated several times, until the user is satisfied with the width of the retrieval set. In the *Burkholderia* example, the user may generalize the initial query as an H at the region level, and

retrieve also the cases indexed by *HHHHHHHHHHHHH* at the gene level (the other siblings of *HHLHHHHHHHHHH* do not index any real case in this specific situation). The cases indexed by *HHHHHHHHHHHHH* can thus be listed, clarifying that their distance from the original query is greater than zero.

Considering the performance of the Case Based Reasoning module, tests have been conducted on databases containing different number of cases. On the left side of Table 1, we report the time elapsed to generate the multi-dimensional indexing structure from the similarity levels generated by BLAST and properly abstracted. The creation times span from 39 seconds to index 2000 cases, to 163 seconds to index 8000 cases. Even if the creation of the structure takes some time, it is necessary to perform this operation only when a new database is installed (or when a significant number of new cases is stored); then the flexible and efficient query mechanism can start running. The right side of Table 1 shows the time elapsed to perform a query, which spans from few milliseconds to query on 2000 cases, to less than one second to query on 8000 cases. These experiments were conducted on an Intel Core 2 Duo T9400 processor running at 2.53 GHz, equipped with 4 Gb of DDR2 RAM.

Table 1. Execution times to build the multi-dimensional orthogonal index (left) and to execute a query (right)

Multidimensional index structure generation from BLAST		Query execution times with multidimensional index	
N. of cases	Structure generation time (s)	N. of cases	Query execution times (s)
2.000	38,969	2.000	0,138
4.000	80,667	4.000	0,333
6.000	121,618	6.000	0,650
8.000	162,241	8.000	0,905

Protein Domains Mining. Beside the investigation of the biological connection at the gene level using the indexing approach, we are able to exploit the *cases* deriving from the case representation to extract new analogies among nucleotide sequences. In details, we query the Chado database to extract all the protein sequences from the obtained *cases*. Then, we use the co-clustering modules to study the domain/motif composition of protein sequences. As it is well known, the modular nature of proteins shows many advantages: it provides an increased stability and new cooperative functions. The usage of protein domains in the determination of the proteins functions has become essential. Several web applications (e.g. Pfam [17], SMART [25], Interpro [31]) are available to provide an overview of the domain architecture of a polypeptide sequence, and the functions that these domains are likely to perform.

Even though the cited tools allow one to submit a set of protein sequences as input, they perform the domain analysis considering each sequence as a single entity. As a consequence, the user can obtain only a *local* view of the domain

composition, instead of a *global* view, that may emphasize the domains characterizing the entire proteins set.

This fact suggests the need of an automatic tool that offers the possibility to manage the results in order to highlight the association between domains and proteins. For this purpose the BIOBITS system includes a *de novo* algorithm [13]. It allows the simultaneous association between protein sequences and domains/motifs. In this way we are able to identify a richer set of motifs, each one possibly characterizing only some of the sequences in the whole dataset. The algorithm relies on three steps. First, we generate a prefix tree starting from the sequences in the input dataset. This data structure enables the fast extraction of all the frequent domains of length up to a fixed value w. Then, we exploit a constrained co-clustering algorithm [34] in order to find protein domain classes and the associated protein groups. Finally, we associate the obtained clusters by means of a statistical measure. This measure individuates for each domain cluster the corresponding protein cluster containing it. The statistical measure can associate some protein clusters to any domain cluster, or some protein cluster to more than one domain cluster.

In the presented case study we consider a domain as frequent if it is found at least in the 10% of the sequences given as input, and we set the maximum domain length w equal to 15. The dataset matrix X (defined in Section 4.2) is built using the frequency values stored in the prefix tree. In the definition of the co-clustering constraints, we exploit the Levenshtein distance between two strings. Specifically, we set a must-link constraint on every pair of domains having a distance less than 2. With this limitation, we consider a must-link between two motifs that require only two string operations (i.e. insertion, deletion or substitution) to transform one motif into the other. Otherwise, all the pairs that match by at most two characters are subject to a cannot-link constraint. The stop condition of the co-clustering algorithm is set to be $\tau = 10^{-3}$.

With the above described experimental setting, we performed two types of experiments. In the first experiment we compose the set of input sequences by combing all the *Burkholderia xenovorans*'s protein sequences of genes belonging to *DCW* cluster, stored in table *Protein* of Chado database. The aim of this experiment is the identification of the protein domains common to a *DCW cluster* gene subfamily. We obtain the six motifs reported in logos representation in Figure 8: panel (a) shows the sequence logo representation of the two domains associated to the fts gene family while panel (b) reports the sequence logo representation of the four domains associated to mur gene family.

In order to validate the reliability of our approach we compare our results with respect to the biological knowledge reported in the review by Clyde A. Smith [36]. Smith describes the three domain architectures characterizing the mur ligases. Two of these domains have essentially conserved topology. The author deeply studied the motif composition of one domain, ATPase. It is characterized by a small number of essential structural motifs that include the P-loop motif. The sequence comparisons reported by Smith show the strong conservation of P-loop motif in all four mur ligases. From our analysis we obtain two motifs strictly

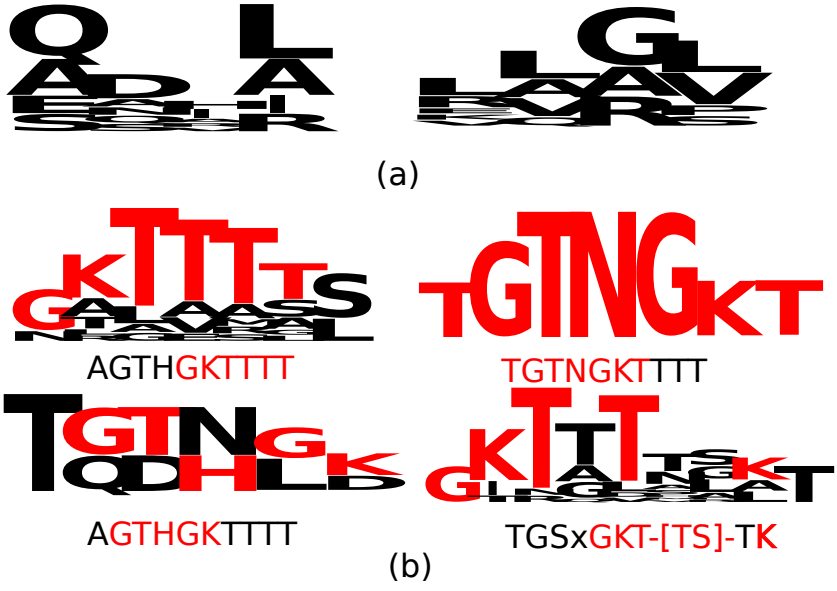

Fig. 8. Sequence logo representation of the motifs obtained by the co-clustering module on *Burkholderia xenovorans*'s *DCW cluster* protein sequences

related to the mur subfamily: in Figure 8(b) we highlight the residues common to the Smith's consensus sequences.

In the second experiment, we exploit the *Phylogenetic classification* table joined to the *Protein* table stored in Chado. The purpose of the second experiment is to extend our analysis to other species of *Burkholderia*. In detail, we single out 13 species:
Burkholderia cepacia, *Burkholderia ambifaria*, *Burkholderia cenocepacia*, *Burkholderia multivorans*, *Burkholderia phytofirmans*, *Burkholderia vietnamiensis*, *Burkholderia glumae Burkholderia xenovorans*, *Burkholderia dolosa*, *Burkholderia graminis*, *Burkholderia phymatum*, *Burkholderia rhizoxinica* and *Burkholderia ubonensis*.

The focus of the task is extending the previous association motif/gene subfamily to all *Burkholderia* genus. This kind of analysis is linked to the possibility of understanding if there are one or more domains joined between different species. Figure 9 shows the co-clusters obtained for genes murC, murD, and murE. Our findings confirm that the gene subfamilies are associated with at least one motif and this association is shared with all the orthologous sequences in the *Burkholderia*'s species. The founded domains lead to identify homologous genes, that may catch the evolutionary relationship among a set of genus. The new pieces of information are then stored in the Chado database.

Fig. 9. Co-clusters obtained by performing the co-clustering module on a set of *Burkholderia* species' *DCW cluster* protein sequences

6 Conclusions

In this paper we reported on the on-going BIOBITS project whose goal is to extensively develop a computational genomic comparison (known as syntheny) focused on the *Ca. Glomeribacter gigasporarum bacterium* and *arbuscular mycorrhiza* fungi genome.

We presented the software architecture essentially developed over an existing software layer provided by GMOD Community. GMOD system offers powerful data visualization and analysis tools, data warehouse modules, such as BioMart and the possibility to exploit import modules for the inclusion of data from the external, public resources. Furthermore, it contains the Chado database which presents an extensible and flexible model for any organism species built upon the generic concept of feature which can be customized by the use of types and ontologies.

We presented the logical data representation of the genomic and proteomic components of the biological problem: it has the form of a double star schema - the first one centered around the genetic fragments composing the genome and the second one on the proteins encoded by the genes.

Then, we describe the main software blocks of BIOBITS system: a Case-Based Reasoning module and a co-clustering module, which allow the user to retrieve and analyse in a flexible and intelligent way the data coming from the multidimensional star schema. Both these modules complement each other. Case-Based Reasoning and temporal analysis retrieve the information at different abstraction levels, as needed by the analyst. Co-clustering provides a novel information to genetic sequences based on computational data mining algorithms.

In the last part of the paper, we describe a case study showing how these modules inter-operate to provide new information. Interesting results have been obtained, with a confirmation from other research studies. The confirmed reliability of our approach encourages us to continue our research on the endosymbiont bacterium *Candidatus* Glomeribacter gigasporarum.

References

1. Acedb, `http://www.acedb.org/`
2. Aamodt, A., Plaza, E.: Case-Based Reasoning: foundational issues, methodological variations and systems approaches. AI Communications 7, 39–59 (1994)
3. Altschul, S., Gish, W., Miller, W., Myers, E., Lipman, D.: Basic local alignment search tool. J. Mol. Biol. 215, 403–410 (1990)
4. Bakker, H., Cummings, C., Ferreira, V., Vatta, P., Orsi, R., Degoriciden ja, L., Barker, M., Petrauskene, O., Furtado, M., Wiedmann, M.: Comparative genomics of the bacterial genus Listeria: Genome evolution is characterized by limited gene acquisition and limited gene loss. BMC Genomics 11 (2010)
5. Bellazzi, R., Larizza, C., Riva, A.: Temporal abstractions for interpreting diabetic patients monitoring data. Intelligent Data Analysis 2, 97–122 (1998)
6. Bentley, S., Parkhill, J.: Comparative genomic structure of prokaryotes. Annual Review of Genetics 38, 771–792 (2004)
7. Bianciotto, V., Lumini, E., Bonfante, P., Vandamme, P.: Candidatus Glomeribacter gigasporarum, an endosymbiont of arbuscular mycorrhizal fungi. Int. J. Syst. Evol. Microbiol. 53, 121–124 (2003)
8. BioMart (2003), `http://www.biomart.org/`
9. Bonfante, P., Anca, I.: Plants, Mycorrhizal Fungi, and Bacteria: A Network of Interactions. Annu. Rev. Microbiol. 63, 363–383 (2009)
10. Carvalho, F., Souza, R., Barcellos, F., Hungria, M., Vasconcelos, A.: Genomic and evolutionary comparisons of diazotrophic and pathogenic bacteria of the order Rhizobiales. BMC Microbiology 10, 1–12 (2010)
11. Commins, J., Toft, C., Fares, M.: Computational Biology Methods and Their Application to the Comparative Genomics of Endocellular Symbiotic Bacteria of Insects. Biomedical Procedures Online 11, 52–78 (2009)
12. Cordero, F., Ghignone, S., Lanfranco, L., Leonardi, G., Meo, R., Montani, S., Roversi, L.: BIOBITS: A Study on Candidatus Glomeribacter Gigasporarum with a Data Warehouse. In: Bohm, C. (ed.) Database Technology for Life Sciences and Medicine Claudia Plant, ch. 10, pp. 203–220 (2011)
13. Cordero, F., Visconti, A., Botta, M.: A new protein motif extraction framework based on constrained co-clustering. In: Proceedings of the 24th Annual ACM Symposium on Applied Computing, pp. 776–781 (2009)
14. Dhillon, I., Mallela, S., Modha, D.: Information-theoretic co-clustering. In: Proceedings ACM SIGKDD 2003, pp. 89–98 (2003)

15. Eilbeck, K., Lewis, S.: Sequence Ontology Annotation Guide. Computational Functional Genomics 5(8), 642–647 (2004)
16. Field, D., Wilson, G., van der Gast, C.: How do we compare hundreds of bacterial genomes? Current Opinion in Microbiology 9, 499–504 (2006)
17. Finn, R., Mistry, J., Schuster-Bckler, B., Griffiths-Jones, S., Hollich, V., Lassmann, T., Moxon, S., Marshall, M., Khanna, A., Durbin, R., Eddy, S., Sonnhammer, E., Bateman, A.: Pfam: clans, web tools and services. Nucleic Acids Res. 34, 247–251 (2006)
18. Gao, B., Mohan, R., Gupta, R.: Phylogenomics and protein signatures elucidating the evolutionary relationships among the Gammaproteobacteria. International Journal of Systematic and Evolutionary Microbiology 59, 234–247 (2009)
19. GenBank (2000), http://www.ncbi.nlm.nih.gov/Genbank/
20. Han, J., Kamber, M.: Data Mining, Concepts and techniques. Academic press, London (2001)
21. Hu, J., et al.: The ARKdb: genome databases for farmed and other animals. Nucleic Acids Res. 29, 106–110 (2001)
22. Hulo, N., Bairoch, A., Bulliard, V., Cerutti, L., Castro, E.D., Langendijk-genevaux, P., Pagni, M., Sigrist, C.: The prosite database. Nucleic Acids Res. 34, 227–230 (2006)
23. Kersey, P.J., Lawson, D., Birney, E., Derwent, P.S., Haimel, M., Herrero, J., Keenan, S., Kerhornou, A., Koscielny, G., Kahari, A., Kinsella, R.J., Kulesha, E., Maheswari, U., Megy, K., Nuhn, M., Proctor, G., Staines, D., Valentin, F., Vilella, A.J., Yates, A.: Ensembl genomes: Extending ensembl across the taxonomic space. Nucleic Acids Research (November 2009), http://dx.doi.org/10.1093/nar/gkp871
24. Lazzarato, F., Franceschinis, G., Botta, M., Cordero, F., Calogero, R.: RRE: a tool for the extraction of non-coding regions surrounding annotated genes from genomic datasets. Bioinformatics 20, 2848–2850 (2004)
25. Letunic, I., Copley, R., Pils, B., Pinkert, S., Schultz, J., Bork, P.: SMART 5: domains in the context of genomes and networks. Nucleic Acids Res. 34, 257–260 (2006)
26. Lumini, E., Ghignone, S., Bianciotto, V., Bonfante, P.: Endobacteria or bacterial endosymbionts? To be or not to be. New Phytol. 170, 205–208 (2006)
27. MAGE Community, MGED Group: MicroArray Gene Expression (MAGE) Project (2000), http://scgap.systemsbiology.net/standards/mage_miame.php
28. Montani, S.: Exploring new roles for case-based reasoning in heterogeneous AI systems for medical decision support. Applied Intelligence 28, 275–285 (2008)
29. Montani, S., Bottrighi, A., Leonardi, G., Portinale, L., Terenziani, P.: Multi-level abstractions and multi-dimensional retrieval of cases with time series features. In: McGinty, L., Wilson, D.C. (eds.) ICCBR 2009. LNCS, vol. 5650, pp. 225–239. Springer, Heidelberg (2009)
30. Moran, N., McCutcheon, A., Nakabachi, P.: Genomics and evolution of heritable bacterial symbionts. Annu. Rev. Genet. 42, 165–190 (2008)
31. Mulder, N., Apweiler, R., Attwood, T., Bairoch, A., Bateman, A., Binns, D., Bork, P., Buillard, V., Cerutti, L., Copley, R., Courcelle, E., Das, U., Daugherty, L., Dibley, M., Finn, R., Fleischmann, W., Gough, J., Haft, D., Hulo, N., Hunter, S., Kahn, D., Kanapin, A., Kejariwal, A., Labarga, A., Langendijk-Genevaux, P., Lonsdale, D., Lopez, R., Letunic, I., Madera, M., Maslen, J., McAnulla, C., McDowall, J., Mistry, J., Mitchell, A., Nikolskaya, A., Orchard, S., Orengo, C., Petryszak, R., Selengut, J., Sigrist, C., Thomas, P., Valentin, F., Wilson, D., Wu, C., Yeats, C.: New developments in the InterPro database. Nucleic Acids Res. 35, 224–228 (2007)

32. Ogier, J., Calteau, A., Forst, S., Goodrich-Blair, H., Roche, D., Rouy, Z., Suen, G., Zumbihl, R., Givaudan, A., Tailliez, P., Medigue, C., Gaudriault, S.: Units of plasticity in bacterial genomes: new insight from the comparative genomics of two bacteria interacting with invertebrates, Photorhabdus and Xenorhabdus. BMC Genomics 11, 1–10 (2010)
33. Osborne, B.: GMOD Community: GMOD (2000), `http://gmod.org/wiki/Main_Page`
34. Pensa, R., Boulicaut, J.F., Cordero, F., Atzori, M.: Co-clustering Numerical Data under User-defined Constraints. Statistical Analysis and Data Mining (2010)
35. Santoni, D., Romano-Spica, V.: Comparative genomic analysis by microbial COGs self-attraction rate. Journal of Theoretical Biology 258, 513–520 (2009)
36. Smith, C.A.: Structure, Function and Dynamics in the mur Family of Bacterial Cell Wall Ligases. Journal of Molecular Biology 362, 640–655 (2006)
37. Thomas, T., Rusch, D., DeMaere, M., Yung, P., Lewis, M., Halpern, A., Heidelberg, K., Egan, S., Steinberg, P., Kjelleberg, S.: Functional genomic signatures of sponge bacteria reveal unique and shared features of symbiosis. ISME Journal 4, 1557–1567 (2010)
38. Watson, I.: Applying Case-Based Reasoning: techniques for enterprise systems. Morgan Kaufmann, San Francisco (1997)
39. Xu, Y.: Computational Challenges in Deciphering Genomic Structures of Bacteria. Journal of Computer Science and Technology 25, 53–73 (2009)
40. Shahar, Y.: A framework for knowledge-based temporal abstractions. Artificial Intelligence 90, 79–133 (1997)
41. Zucko, J., Dunlap, W., Shick, J., Cullum, J., Cercelet, F., Amin, B., Hammen, L., Lau, T., Williams, J., Hranueli, D., Long, P.: Global genome analysis of the shikimic acid pathway reveals greater gene loss in host-associated than in free-living bacteria. BMC Genomics 11 (2010)