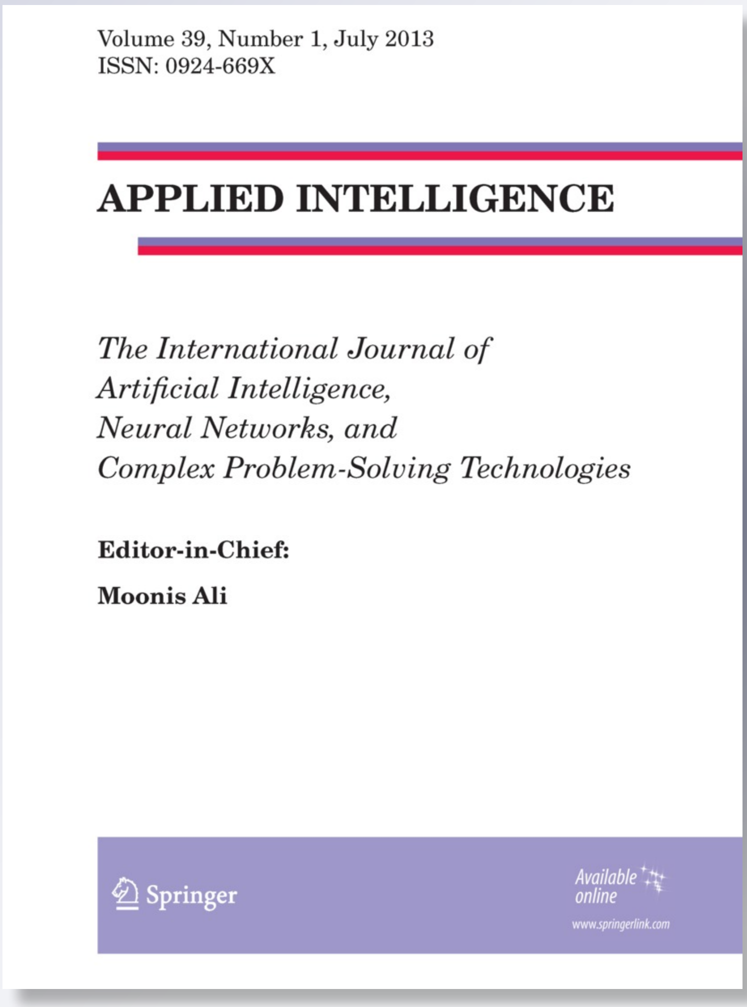
Flexible case-based retrieval for comparative genomics

Stefania Montani, Giorgio Leonardi, Stefano Ghignone & Luisa Lanfranco

Applied Intelligence
The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies

ISSN 0924-669X
Volume 39
Number 1

Appl Intell (2013) 39:144-152
DOI 10.1007/s10489-012-0399-z


Volume 39, Number 1, July 2013
ISSN: 0924-669X

APPLIED INTELLIGENCE

The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies

Editor-in-Chief:

Moonis Ali

Springer

Available online
www.springerlink.com

Springer

Your article is protected by copyright and all rights are held exclusively by Springer Science +Business Media New York. This e-offprint is for personal use only and shall not be self-archived in electronic repositories. If you wish to self-archive your article, please use the accepted manuscript version for posting on your own website. You may further deposit the accepted manuscript version in any repository, provided it is only made publicly available 12 months after official publication or later and provided acknowledgement is given to the original source of publication and a link is inserted to the published article on Springer's website. The link must be accompanied by the following text: "The final publication is available at link.springer.com".

Springer

Author's personal copy

Appl Intell (2013) 39:144–152
DOI 10.1007/s10489-012-0399-z

Flexible case-based retrieval for comparative genomics

**Stefania Montani · Giorgio Leonardi ·
Stefano Ghignone · Luisa Lanfranco**

Published online: 6 December 2012
© Springer Science+Business Media New York 2012

Abstract Comparative genomics represents a key instrument to discover or validate phylogenetic relationships, to give insights on genome evolution, and to infer metabolic functions of a given organism. A tool for properly supporting comparative genomics is of paramount importance in several application domains.

We have developed a tool for supporting customized comparative genomics searches. The tool is based on the retrieval step of the Case-Based Reasoning methodology. It takes advantage of an abstraction technique similar to Temporal Abstractions, thus allowing us to neglect irrelevant details.

By means of our tool, retrieval is made flexible by the use of abstractions, and efficient by the use of proper taxonomic index structures. Moreover, end-users are allowed to progressively relax or refine their queries in an interactive way. The tool functionalities are exemplified referring to the study of endobacteria living in arbuscular mycorrhizal fungi.

S. Montani (✉) · G. Leonardi
DISIT—Computer Science Institute, University of Piemonte Orientale, Alessandria, Italy
e-mail: stefania.montani@unipmn.it

G. Leonardi
e-mail: giorgio.leonardi@unipmn.it

S. Ghignone · L. Lanfranco
IPP—CNR, UAS Torino, Turin, Italy

S. Ghignone
e-mail: stefano.ghignone@unito.it

L. Lanfranco
e-mail: luisa.lanfranco@unito.it

L. Lanfranco
Department of Life Science and Systems Biology, University of Turin, Turin, Italy

Keywords Comparative genomics · Case-Based Reasoning · Flexible retrieval · Index-based retrieval · Interactivity

1 Background

Comparative genomics represents a key instrument to answer fundamental questions concerning the biology, ecology and evolutionary history of an organism, or of a biological system and of its composing elements. In particular, the comparative genomics approach is extremely useful when biochemical and physiological data are not available and/or hard to obtain.

The introduction of massive parallel technologies used in next-generation sequencing (NGS) are revolutionizing genomic research [1]. The manipulation and interpretation of the millions of nucleotide sequences generated by NGS present significant computational challenges from the reads assembly to comparisons of many entire genomes. Therefore, there is urgent need of platforms able to handle such large-scale genomic data.

In this frame, we are developing a modular architecture for comparative genomics. We have chosen to exploit as much as possible the available tools built around GMOD, the Generic Model Organism Database project [2]. Indeed, GMOD brought to the development of a whole, and still under expansion, collection of open source software tools for creating and managing genome-scale biological databases.

In particular, we are extending the functionalities offered by GMOD, in order to properly meet the needs of a specific comparative genomics study we are interested in, within the BIOBITS project (see Sect. 1.1).

In this paper, we will focus on the characteristics of one single extension we are providing, namely a flexible genome

Author's personal copy

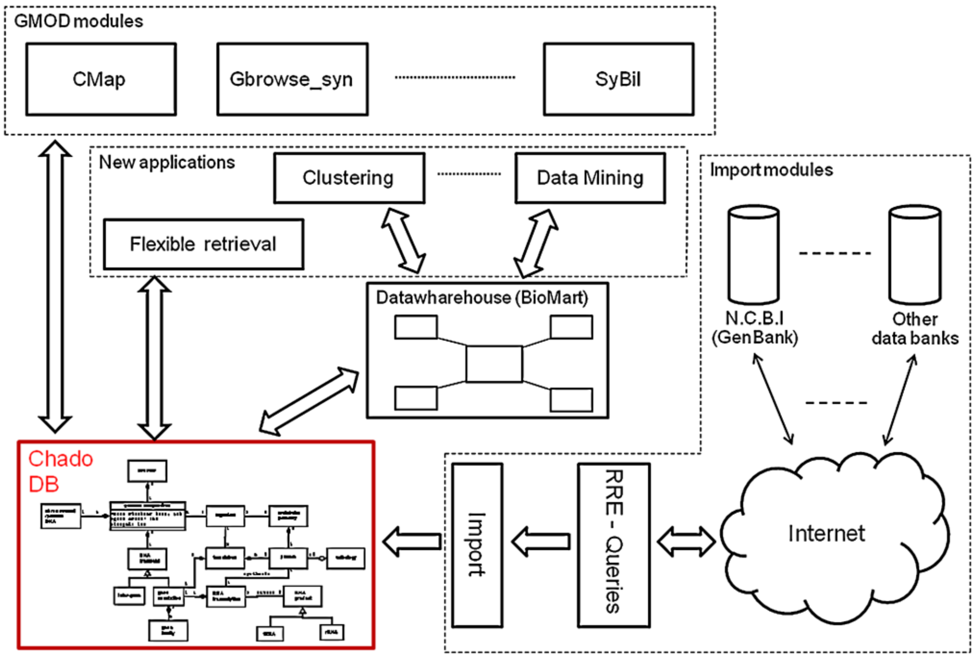


Fig. 1 System architecture

search and retrieval tool, based on the Case-Based Reasoning (CBR) methodology [3]. Such a tool allows us to search the genome database by expressing queries at different levels of abstraction detail, resorting to a technique similar to Temporal Abstractions (TA) [4, 5]. Moreover, end-users are allowed to progressively relax or refine their queries, in an interactive way. Finally, retrieval is made efficient by the use of multi-dimensional orthogonal index structures, which allow for early pruning and focusing.

We will present the tool referring to a case study, in which we compare genomes of bacterial species belonging to the Burkholderiaceae family. This family includes a number of well described free-living species and the still enigmatic obligate endosymbionts such those living in arbuscular mycorrhizal fungi (AMF).

The paper is organized as follows. Section 1.1 quickly describes the application we will refer to in order to illustrate our approach. Section 2 first illustrates the general architecture we are implementing for genome analysis in the project, and then focuses on our retrieval tool. Section 3 presents a case study. Section 4 discusses related works and Sect. 5 summarizes our conclusions.

1.1 The application domain

Bacterial endosymbionts in the animal kingdom have been and are excellent models for investigating important biological events, such as organelle evolution, genome reduction, and transfer of genetic information among host lineages [6]. By contrast, the knowledge on endobacteria living in fungi is limited [7, 8]. Arbuscular mycorrhizal fungi (AMFs) are a crucial component of soil microbial communities and exert positive impacts on plant health and productivity in natural and agricultural systems. They establish a symbiotic association with the root of land plants, providing a better mineral nutrition and an increased tolerance to stress conditions. AMFs are thus a significant resource for sustainable agriculture. AMFs are often also in symbiosis with uncultivable bacteria which are hosted in hyphae of AMFs, leading to a complex tripartite system (i.e. endobacterium-AMF-plant roots) [9, 10].

In our current project BIOBITS (see the Acknowledgments), we focus on the *Candidatus* Glomeribacter gigasporarum [11], endobacterium of the AMF *Gigaspora margarita* (isolate BEG34), currently used as a model system to investigate endobacteria-AMFs interactions.

2 Methods

2.1 System architecture

The system architecture we are developing within the BIOBITS project has been engineered to exploit the standard modules and interfaces offered by the GMOD project [2], and completed with custom modules to provide new functionalities (see Fig. 1).

The main module of the system contains the database, which provides all the data needed to perform the in-silico activities. We adopted the GMOD Chado database schema [12], to take advantage of its completeness and of its support for controlled vocabularies and ontologies. Furthermore, Chado is the standard database for most of the GMOD

Author's personal copy

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. The database in this module stores and provides all the information about the organisms to be studied (mainly bacteria), their genomes, their known annotations, their proteins and metabolic pathways. It also includes the newly discovered annotations, which can be stored and managed locally until they are confirmed and published.

As explained, our database contains information to be used and stored locally, but we have added the possibility to populate and update it with information retrieved from the biological databases accessible through the Internet. This feature is provided by the set of modules in the *Import modules* section (see Fig. 1). The main module (*RRE—Queries*), which is built on the basis of a previously published tool [13], performs queries to different biological databases through the Internet (e.g. the GenBank [14]) and converts the results into a standard format. Afterwards, the *Import module* inserts or updates the retrieved information into the Chado database. This process can be started on-demand or performed automatically on a regular basis in order to keep the local database up-to-date.

Chado also acts as the data interface for the software layers implementing the functionalities and tools used by the researchers. From the architectural point of view, we offer two types of services: the services implemented through existing modules of GMOD (*GMOD modules* section in Fig. 1), and new services implemented through new modules, developed ad-hoc (*New applications* section).

The existing GMOD modules we are exploiting are the following:

- *CMap*, which allows users to view comparisons of genetic and physical maps. The package also includes tools for maintaining map data;
- *GBrowse*, which is a genome viewer, and also permits the manipulation and the display of annotations on genomes;
- *GBrowse_syn*, which is a GBrowse-based synteny browser designed to display multiple genomes, with a central reference species compared to two or more additional species;
- *SyBil*, which is a system for comparative genomics visualizations.

All the GMOD tools exploit a web-based interface to be more user-friendly and easy to use. Moreover, they can be reused as they are, but they can also be customized to meet the researchers' recommendations before being integrated in our software architecture.

As for the new modules, they consist of:

- a set of clustering and other data mining modules (whose description is outside the scope of this paper). Such tools rely on *BioMart* [15], a GMOD facility able to reorganize the information stored in the Chado database into a data warehouse;
- the *flexible retrieval module*, which is the main topic of this paper, and which will be described in Sect. 2.2.

Every new module added in the *New applications* section of our architecture, or every customized module in the *GMOD modules* section, connects to the other modules of our architecture using GMOD standard interfaces. Therefore, each of them can be published to the GMOD community, in order to extend and enrich the functionalities of this platform.

2.2 A flexible retrieval tool

The flexible retrieval module we have designed in the project architecture implements the retrieval step of the Case-Based Reasoning (CBR) [3] 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 after, if necessary, properly
3. *revising* them; the current case can then be
4. *retained* and put into the system knowledge base, called the case base.

Purely retrieval systems, leaving to the user the completion of the reasoning cycle (steps 2 to 4), are however very valuable decision support tools [16], especially when automated adaptation strategies can hardly be identified, as in biology and in medicine [17]. In BIOBITS we are following exactly this research line.

In the flexible retrieval module, the information stored in cases is related to genomes expressed as sequences of nucleotides, each one taken from a different organism, and properly aligned with the genome of a reference organism.

Completing the alignment task is therefore a prerequisite for representing cases in our library.

Details of our alignment strategy, of case representation and of case retrieval are discussed in the next sections.

2.2.1 Case representation

As previously observed, our first step towards case representation requires alignment. To this end, we rely on BLAST [18], a state-of-the-art local alignment algorithm, specifically designed for bioinformatics applications.

From a typical BLAST output (Fig. 2), one can extract basic information (e-value, score and identity percentage) that can be easily plotted as represented in Fig. 3, providing a piecewise constant function, which graphically represents the alignment itself. Specifically, Fig. 3 refers to e-value.

Author's personal copy

Quantitative e-values provided by BLAST can be converted into a set of qualitative levels (e.g. low, high similarity), thus providing a "higher level" view of the information, able to abstract from unnecessary details. To perform such a conversion, we propose a semantic-based abstraction process, similar to the Temporal Abstractions (TA) techniques [4, 5].

TA is an Artificial Intelligence methodology which allows us to move from a point-based to an interval-based representation of a time series, where time series points are converted into intervals (episodes), aggregating adjacent points sharing a common behavior, persistent over time. In particular, state abstractions [5] allow us to extract episodes associated with qualitative levels of the variable represented by the time series. In state abstractions, the mapping between qualitative abstractions and quantitative values has to be parameterized on the basis of domain knowledge.

In our framework, we adopt this methodology with a main difference: the independent variable is the symbol position in the aligned strings, instead of time. The values to be converted into qualitative levels are then the e-values calculated between the two strings themselves. An example is provided in Fig. 3, where e.g. a $10^{-32} <$ e-value $< 10^{-5}$ is considered to be moderately high (Hm).

Indeed, on the basis of domain knowledge, we define a whole taxonomy of qualitative levels, where high values can be further specialized (see Fig. 4). Abstractions of e-values at different levels of detail can support different requests of the researchers.

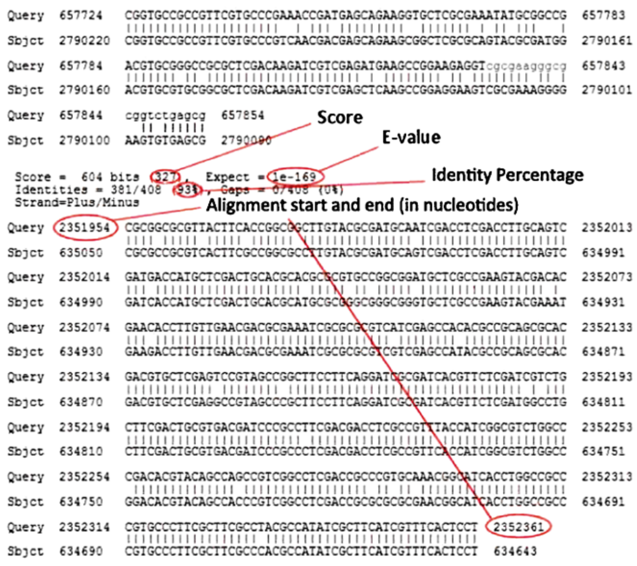
We also allow the abstraction of information about the "localization" of the aligned substrings along the nucleotide sequences at different granularities (see Fig. 5). This allows us to visualize genome information at the level of groups of nucleotides, genes, regions, chromosomes or even complete genomes.[1]

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 difficult to identify at the level of single genes (see the example in Sect. 3).

In particular, as shown in Fig. 5, we abstract the row data at a minimum granularity of 500-nucleotides-long sequences. This choice is motivated by the fact that the average gene length in our application domain is 1000 nucleotides—but our framework is parametric with respect to this choice.

Indeed, the whole taxonomy of granularities is strongly influenced by domain semantics. 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 (see Sect. 2.2.2).

To summarize, in our framework case representation is obtained as follows. First, a pair of nucleotide strings, optimally aligned as calculated by BLAST, is taken. In particular, for each subsequence of nucleotides, the e-value with


```
Query  657724   CGGTGCCGCCGTTCGTGCCCGAAACCGATGAGCAGAAGGTGCTCGCGAAATATGCGGCCG  657783
                |||||||||||||||||||||  | ||| |||||||||  ||||||| | || |||   |
Sbjct  2790220  CGGTGCCGCCGTTCGTGCCCGTCAACGACGAGCAGAAGCGGCTCGCGCAGTACGCGATGG  2790161

Query  657784   ACGTGCGGGCCGCGCTCGACAAGATCGTCGAGATGAAGCCGGAAGAGGTcgcgaagggcg  657843
                ||||||| || ||||||||||||||||||||| | |||||||| || |||||||| || |
Sbjct  2790160  ACGTGCGTGCGGCGCTCGACAAGATCGTCGAGCTCAAGCCGGAGGAAGTCGCGAAAGGGG  2790101

Query  657844   cggtctgagcg  657854
                  || ||||||
Sbjct  2790100  AAGTGTGAGCG  2790090

 Score =  604 bits (327),  Expect = 1e-169
 Identities = 381/408 (93%), Gaps = 0/408 (0%)
 Strand=Plus/Minus

Query  2351954  CGCGGCGCGTTACTTCACCGGCGGCTTGTACGCGATGCAATCGACCTCGACCTTGCAGTC  2352013
                |||| ||||| ||||| |||||| ||||||||||||||| ||||||||||||||||||||
Sbjct  635050   CGCGCCGCGTCACTTCGCCGGCGCCTTGTACGCGATGCAGTCGACCTCGACCTTGCAGTC  634991

Query  2352014  GATGACCATGCTCGACTGCACGCACGCGCGTGCCGGCGGATGCTCGCCGAAGTACGACAC  2352073
                ||| |||||||||||||||||||| ||||| || ||||| ||||||||||||||||| |
Sbjct  634990   GATCACCATGCTCGACTGCACGCATGCGCGGGCGGGCGGGTGCTCGCCGAAGTACGAAAT  634931

Query  2352074  GAACACCTTGTTGAACGACGCGAAATCGCGCGTCATCGAGCCACACGCCGCAGCGCAC  2352133
                ||| |||||||||||||||||||||||||||||| |||||||| ||||||||||||||
Sbjct  634930   GAAGACCTTGTTGAACGACGCGAAATCGCGCGTCGTCGAGCCATACGCCGCAGCGCAC  634871

Query  2352134  GACGTGCTCGAGTCCGTAGCCGGCTTCCTTCAGGATCGCGATCACGTTCTCGATCGTCTG  2352193
                |||||||||||| |||||||| |||||||||||||||||||||||||||||||| | |||
Sbjct  634870   GACGTGCTCGAGGCCGTAGCCCGCTTCCTTCAGGATCGCGATCACGTTCTCGATGGCCTG  634811

Query  2352194  CTTCGACTGCGTGACGATCCCGCCTTCGACGACCTCGCCGTTTACCATCGGCGTCTGGCC  2352253
                |||||||||||||||||||||||| ||||||||||||||||| |||||||||||||||||
Sbjct  634810   CTTCGACTGCGTGACGATCCCGCCCTCGACGACCTCGCCGTTCACCATCGGCGTCTGGCC  634751

Query  2352254  CGACACGTACAGCCAGCCGTCGGCCTCGACCGCCCGTGCAAACGGCATCACCTGGCCGCC  2352313
                 |||||||||||||| ||||||||||||||||| || || ||||||||||||||||||||
Sbjct  634750   GGACACGTACAGCCACCCGTCGGCCTCGACCGCGCGCGCGAACGGCATCACCTGGCCGCC  634691

Query  2352314  CGTGCCCTTCGCTTCGCCTACGCCATATCGCTTCATCGTTTCACTCCT  2352361
                |||||||||||||||||| |||||||||||||||||||||||||||||
Sbjct  634690   CGTGCCCTTCGCTTCGCCCACGCCATATCGCTTCATCGTTTCACTCCT  634643
```

Fig. 2 BLAST sequence alignment

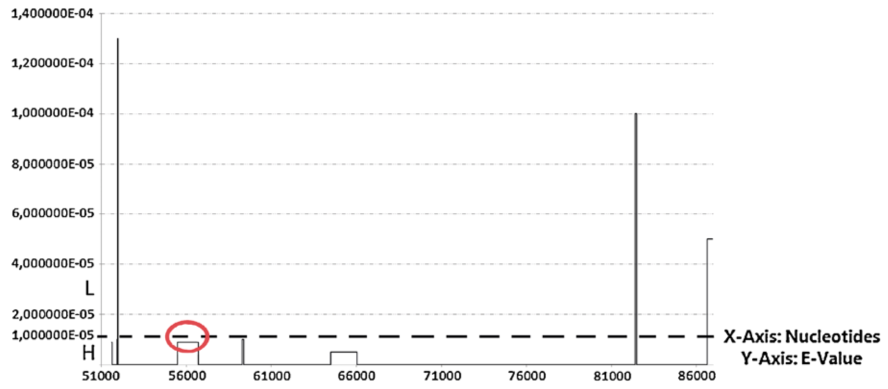


Fig. 3 A graphical visualization of sequence alignment (*x-axis*: nucleotide position of the alignment with respect to the reference string; *y-axis*: e-values)

[1]Since, in our application, the full genome of an organism is typically subdivided into one or more chromosomes/plasmids, similarity between two genomes has to be calculated by applying an aggregation operation to the similarities at the single chromosome/plasmid level. We are currently using the arithmetic mean as an aggregation operator.

Author's personal copy

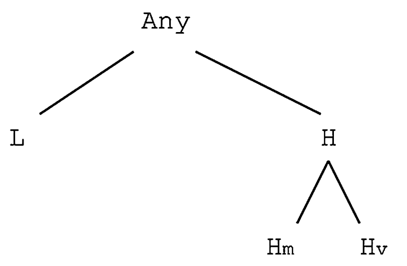


Fig. 4 Taxonomy of state abstraction symbols. The qualitative symbol low (L) corresponds to e-value $> 10^{-5}$, while high (H) corresponds to e-value $\leq 10^{-5}$. Further, very high (Hv) corresponds to e-value $< 10^{-32}$; moderately high (Hm) corresponds to $10^{-32} \leq$ e-value $\leq 10^{-5}$

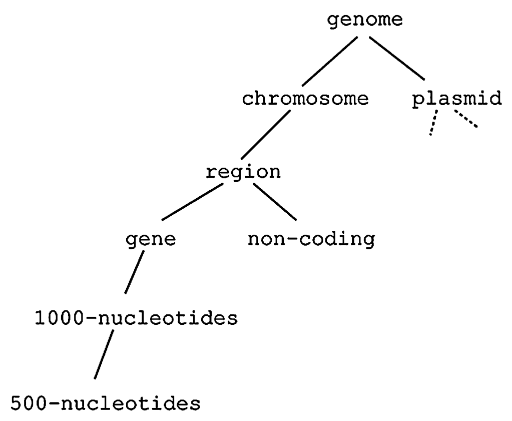


Fig. 5 Taxonomy of sequence granularities. Observe that, given the lowest level of granularity, corresponding to 500-nucleotides-length sequences, it is possible to move up both to 1000-nucleotides-length sequences (and additional levels could be added), and/or, resorting to annotations, to the gene level

respect to the aligned nucleotides in the paired string is provided. Abstractions on such quantitative levels are then calculated, and allow to convert these values into qualitative ones. 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 Fig. 5). The resulting string of symbols is finally 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 coarser levels in both dimensions (i.e. in the dimension of the taxonomy of symbols, and in the one of granularities). Such a conversion is actually the means by which we support flexible case retrieval, and will be described in the next section.

2.2.2 *Case retrieval*

As described in Sect. 2.2.1, our retrieval framework allows for abstractions in two dimensions, namely a taxonomy of state abstraction symbols, and a variety of granularities.

Taking advantage of this data representation, we support flexible retrieval.

In particular, we allow users to express their queries at any level of detail, both in the dimension of the taxonomy of symbols and in the dimension of granularity. Obviously, since cases are stored at the ground level in both dimensions, in order to identify the cases that match a specific query, we have to provide a function for scaling up (called "up" henceforth) two or more symbols expressed at a specific granularity level to a single symbol expressed at a coarser one. Moreover, we need to define a proper distance function for retrieval.

The data structures described in Sect. 2.2.1, as well as the *up* and the distance functions, have to be detailed on the basis of the semantics of the specific application domain.

In particular, scaling up between the lower levels in the taxonomy of granularities requires the adoption of specific domain-dependent choices. For instance, in our domain it makes sense that an H (high) and an L (low) symbols at the 500-nucleotides-length level are converted into an H symbol at the 1000-nucleotides-length one. Scaling up e.g from the nucleotides level to the genes one is even more application-dependent, and requires to parse and exploit annotations. Annotations, in fact, allow us to know where the gene is located on the nucleotide sequence. In this case, the *up* function also has to specify how to scale up the two or more symbols in the 500-nucleotides-length sequences intersecting the gene to the single symbol labeling the gene itself. Similar issues have to be dealt with when scaling up to regions.

However, the framework is parametric with respect to these choices, and can be quickly adapted to different domains.

Moreover, we have identified a set of general (domain independent) "consistency" constraints, that any meaningful choice must satisfy, in order to avoid ambiguous or meaningless situations. For instance, distance "preserves" ordering also in case *isa* relationships between symbols are involved (e.g. the distance between L (low) and Hm (moderately high) is smaller than the distance between L (low) and Hv (very high)). The exhaustive presentation of such constraints can be found in [19].

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 described in Sect. 2.2.1. 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", see Fig. 4) 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

Author's personal copy

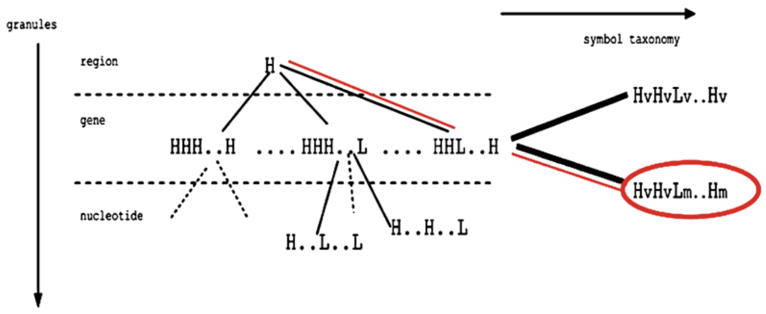
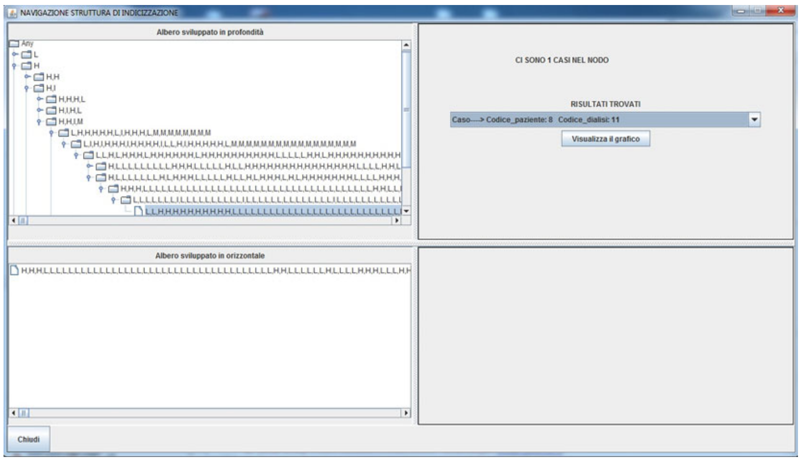


Fig. 6 An example multi-dimensional orthogonal index


NAVIGAZIONE STRUTTURA DI INDICIZZAZIONE
Albero sviluppato in profondità
Albero sviluppato in orizzontale
Visualizza il grafico
Chiudi

```
Any
  L
  H
    H,H
    H,I
      H,H,H,L
      H,I,H,L
      H,H,I,M
        L,H,H,H,H,H,L,I,H,H,H,L,M,M,M,M,M,M,M,M
          L,I,H,I,H,H,H,I,H,H,H,H,I,L,L,H,I,H,H,H,H,H,L,M,M,M,M,M,M,M,M,M,M,M,M,M,M,M,M,M
            L,L,H,L,H,H,H,L,H,H,H,H,H,H,L,H,H,H,H,H,H,H,H,H,H,L,L,L,L,L,H,H,L,H,H,H,H,H,H,H,H,H
              H,L,L,L,L,L,L,L,L,L,H,H,H,L,L,L,L,L,H,L,L,H,H,H,H,H,H,H,H,H,H,H,H,H,H,L,L,L,L,H,H,L
              H,L,L,L,L,L,L,L,H,L,H,H,H,L,L,L,L,L,H,L,L,H,L,H,H,H,L,H,L,H,H,H,H,H,H,H,L,L,L,L,H,H,H,
                H,H,H,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,H,H,L,L
                  L,L,L,L,L,L,L,I,L,L,L,L,L,L,L,L,L,I,L,L,L,L,L,L,L,L,L,L,L,L,I,L,L,L,L,L,L,L,L,L
                    L,L,H,H,H,H,H,H,H,H,H,H,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L

H,H,H,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,L,H,H,L,L,L,L,L,L,H,L,L,L,L,H,H,H,L,L,L,H,H
```

CI SONO 1 CASI NEL NODO

RISULTATI TROVATI

Caso----> Codice_paziente: 8 Codice_dialisi: 11

Fig. 7 A snapshot of index navigation as rendered by the system graphical interface

Fig. 6. 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 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 is also possible.

Technically speaking, to answer a query, in order to enter the more proper index structure, we first progressively generalize the query along the secondary dimension (i.e. the symbol taxonomy), while keeping the leading dimension (i.e. granularity) 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 original query leading dimension level. 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 memorization 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 low as possible.

Flexibility and interactivity are also supported by a user-friendly graphical interface, which has been designed by following software engineering principles, in order to enhance usability and user friendliness in the interaction with the system. Through the interface, we provide a graphical representation of the indexes (conceptually depicted as in Fig. 6), whose nodes can be exploded or iconified, facilitating index navigation (see Fig. 7). Moreover, the graphical interface can support the user in selecting the proper navigation direction, providing him/her with quantitative and qualitative information about the cases indexed by sons and siblings of the currently visited node. For instance, we provide information about the number of indexed cases, the sequence of abstractions representing the cases, and the distance from the sequence of abstractions representing the node currently visited by the user.

Very encouraging experimental results have already been obtained by resorting to the same framework, in the field of haemodialysis [19].

Author's personal copy

3 Results

In this section we illustrate the retrieval mechanisms of our approach by means of a specific case study. We take as a reference organism a bacterium belonging to the Burkholderia genus. All bacteria belonging to this family share a region, called DCW cluster, which is involved in the synthesis of peptidoglycan precursors and cell division. The DCW cluster is composed by 14 genes: FtsA, FtsI, FtsL, FtsQ, FtsW, FtsZ, mraW, mraY, mraZ, murC, murD, murE, murF, and murG. The prominent feature of this cluster is that it is conserved with a high similarity in many bacterial genomes over a broad taxonomic range. Notwithstanding, some bacteria belonging to the studied family simply miss one of the 14 genes (specifically the third), while all of the others maintain a high similarity at the DCW region level with their relatives. Therefore, it makes sense to define the *up* function as follows: up(HHLHHHHHHHHHHH)=H (where the absence of a gene is identified by a low similarity value in the gene position).

Suppose that, more precisely, the user expresses the query HvHvLvHvHvHvHvHvHvHvHvHvHvHv, aiming at retrieving the specific bacteria missing the third gene, but very similar to the reference one as regards the other genes.

Our system will first generalize the query in the symbol taxonomy dimension, providing the string HHLHHHHHHHHHHH (see Fig. 6), and then in the granularity dimension, providing the query H at the region level. This allows us to enter the index in Fig. 6 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. The index search steps are highlighted in the figure.

Interactive and progressive query relaxation (or refinement) are supported as well in our framework. 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 HHHHHHHHHHHHHH at the gene level (the other siblings of HHLHHHHHHHHHHH do not index any real case in this specific situation). The cases indexed by HHHHHHHHHHHHHH can thus be listed, clarifying that their distance (calculated by any distance function which satisfies the constraints illustrated in [19]) from the original query is greater than zero.

4 Discussion

In 1993, Aaronson [21] suggested that analogical reasoning (which includes CBR) is particularly applicable to the biological domain because biological systems are often homologous, and because biologists often design and perform experiments based on the similarity between features of the new system to be investigated, and already known ones. As a matter of fact, since then, some CBR applications in biology and bioinformatics have been published. The paper in [22] is an interesting survey on the topic. The surveyed papers are mostly related to experimental design in protein crystallization and protein structure prediction. However, one contribution [21] also makes the hypothesis of using a CBR approach for predicting unknown regulatory regions. More recently, a hybrid method (resorting to Bayesian techniques and CBR) for feature selection in microarray data analysis has been presented [23]. Except for the work in [21], however, we are not aware of CBR works in genomic comparison. Moreover, the work in [21] does not support any flexible and interactive case retrieval, as we are able to do by means of the abstraction mechanism.

As stated in Sect. 2.2.1, our abstraction mechanisms resemble the abstraction mechanisms of TA [4, 5]. In fact, the present work has been developed starting from our previous experience on TA-based time series retrieval [19, 20]. With respect to those works, here we have properly adapted the characteristics of the existing framework to the biological domain. Actually, such a framework was designed in a modular and domain-independent way. We have realized the adaptation to the biological domain by acquiring the specific domain knowledge, which is the basis for a proper definition of the taxonomies and of the distance and *up* functions.

As regards TA, they have been extensively utilized in the literature, especially in the medical field, from diabetes mellitus [24, 25], to artificial ventilation of intensive care patients [26] (see also the survey in [27]). However, typically they were exploited with the aim to solve a data interpretation task [4], and not to support flexible retrieval.

The goal of our proposal is to try to fill this gap, by exploiting an abstraction mechanism for supporting data interpretation, as well as case exploration and retrieval; this idea thus appears to be significantly innovative in the recent literature panorama.

As previously observed, one of our main goals is also interactivity. It is worth noting that, in classical CBR systems, interactivity is typically not supported: the user is asked to input the entire, precise problem description as a query for case retrieval. This means s/he must know the relevance of every case feature—which is not always straightforward in practice. A research direction meant to overcome this limitation indeed exists in the CBR literature, and is known as Conversational CBR (CCBR) (see e.g. [28–30]). In CCBR, the user is allowed to input just a brief free text description of the case, to start. The system then supports a progressive query refinement through a conversation, in which best matching cases are listed, and further questions meant to reduce and specialize the retrieval set are asked by the system. Our framework thus loosely resembles CCBR. However,

Author's personal copy

CCBR is characterized by some strong challenges, mainly related to case authoring, and dialog inference. Both aspects are non-trivial, and should be solved by specific modules (based e.g. on machine learning [30] or model-based reasoning [31] techniques). Such an additional effort is not required in our framework.

Interestingly, a number of (non CBR) tools to support bioinformatics applications are available in the literature (see e.g. [32–40]). Most of them are only loosely related to our work. On the other hand, the approaches in [32, 39, 40] deal with comparative genomics. In particular, the works in [32, 40] afford the problem of multiple sequence alignment, also discussed in [41]. The work in [32] is a methodological contribution that introduces a genetic algorithm to explore the search space for the multiple sequence alignment task. The approach also refines the search through local search optimization. The work in [40] describes VISTA, a tool which allows the visualization of pre-computed pairwise and multiple alignments of whole genome assemblies. With respect to these approaches, our tool provides additional interesting features. Namely, it allows us 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, queries can be performed efficiently, and potentially on very large databases.

As a last remark, the work in [33] introduces a technique for pattern matching with regular expressions, which has been tested on biological applications. Indeed, as a future work, we would like to extend our querying capabilities, including the definition of a more powerful query language. Such a language could involve the use of regular expressions (see also [19]). Indeed, queries with regular expressions could capture in an abstract and concise way a set of specific "ground" queries. Therefore, works like the one in [33] will be the object of our study in the future.

5 Conclusions

In this paper, we have described a modular architecture for supporting in-silico comparative genomics analysis, being developed within the BIOBITS project. In particular, we have focused on the main features of a genome search tool, which implements the retrieval step of the CBR cycle. Such a tool provides researchers with flexible retrieval capabilities, in an interactive fashion. Flexibility and interactivity are also supported by a user-friendly graphical interface. Moreover, retrieval performances are optimized by resorting to multi-dimensional orthogonal index structures, allowing for quick query answering.

Our future work will be devoted to prove this statement, by means of an extensive experimental work. Future enhancements, such as the definition of a richer query language, are also foreseen.

Acknowledgements *Funding*: The work is supported by the BIOBITS project, a grant of Regione Piemonte, under the Converging Technologies Call, which involves the University of Turin, the University of Piemonte Orientale, the IPP-CNR and the companies ISAGRO Ricerca s.r.l., GEOL Sas, Etica s.r.l.

References

1. Zhang J, Chiodini R, Badr A, Zhang G (2011) The impact of next-generation sequencing on genomics. J Genet Genomics 38:95–109
2. Osborne B, GMOD Community (2000) GMOD
3. Aamodt A, Plaza E (1994) Case-based reasoning: foundational issues, methodological variations and systems approaches. AI Commun 7:3959
4. Shahar Y (1997) A framework for knowledge-based temporal abstractions. Artif Intell 90:79133
5. Bellazzi R, Larizza C, Riva A (1998) Temporal abstractions for interpreting diabetic patients monitoring data. Intell Data Anal 2:97122
6. Moran NA, McCutcheon AJ, Nakabachi P (2008) Genomics and evolution of heritable bacterial symbionts. Annu Rev Genet 42:165190
7. Lumini E, Ghignone S, Bianciotto V, Bonfante P (2006) Endobacteria or bacterial endosymbionts? To be or not to be. New Phytol 170:205208
8. Lackner G, Moebius N, Partida-Martinez L, Hertweck C (2011) Complete genome sequence of Burkholderia rhizoxinica, an endosymbiont of Rhizopus microsporus. J Bacteriol 193:783–784.
9. Bonfante P, Anca IA (2009) Plants mycorrhizal fungi, and bacteria: a network of interactions. Annu Rev Microbiol 63:363–383
10. Naumann M, Schusler A, Bonfante P (2010) The obligate endobacteria of arbuscular mycorrhizal fungi are ancient heritable components related to the mollicutes. ISME J 4:862–871
11. Ghignone S, Salvioli A, Anca I, Lumini E, Ortu G, Petiti L, Cruveiller S, Bianciotto V, Piffanelli P, Lanfranco L, Bonfante P (2012) The genome of the obligate endobacterium of an AM fungus reveals an interphylum network of nutritional interactions. ISME J 6:136–145
12. Mungall CJ, Emmert DB, The FlyBase Consortium (2007) A Chado case study: an ontology-based modular schema for representing genome-associated biological information. Bioinformatics 23(13):i33746
13. Lazzarato F, Franceschinis G, Botta M, Cordero F, Calogero R (2004) RRE: a tool for the extraction of non-coding regions surrounding annotated genes from genomic datasets. Bioinformatics 20(16):2848–2850
14. http://www.ncbi.nlm.nih.gov/Genbank/
15. http://www.biomart.org/
16. Watson I (1997) Applying case-based reasoning: techniques for enterprise systems. Morgan Kaufmann, San Mateo
17. Montani S (2008) Exploring new roles for case-based reasoning in heterogeneous AI systems for medical decision support. Appl Intell 28:275285
18. http://blast.ncbi.nlm.nih.gov/Blast.cgi
19. Montani S, Leonardi G, Bottrighi A, Portinale L, Terenziani P (2012) Supporting flexible, efficient and user-interpretable retrieval of similar time series. IEEE Trans Knowl Data Eng. doi:10.1109/TKDE.2011.264

Author's personal copy

20. Bottrighi A, Leonardi G, Montani S, Portinale L, Terenziani P (2010) Intelligent data interpretation and case base exploration through temporal abstractions. In: Bichindaritz I, Montani S (eds) Proc international conference on case-based reasoning (ICCBR). LNCS, vol 6176. Springer, Berlin, pp 36–50
21. Aaronson JS, Juergen G, Overton GC (1993) Knowledge discovery in GENBANK. In: Hunter L, Searls D, Shavlik U (eds) Proc international conference on intelligent systems for molecular biology. AAAI Press, Melno Park, pp 3–11
22. Jurisica I, Glasgow J (2004) Applications of case-based reasoning in molecular biology. AI Mag 25(1):85–95
23. Bichindaritz I, Annest A (2010) Case based reasoning with Bayesian model averaging: an improved method for survival analysis on microarray data. In: Bichindaritz I, Montani S (eds) Proc international conference on case-based reasoning (ICCBR). LNCS, vol 6176. Springer, Berlin, pp 346–359
24. Shahar Y, Musen M (1996) Knowledge-based temporal abstraction in clinical domains. Artif Intell Med 8:267–298
25. Bellazzi R, Larizza C, Magni P, Montani S, Stefanelli M (2000) Intelligent analysis of clinical time series: an application in the diabetes mellitus domain. Artif Intell Med 20:37–57
26. Miksch S, Horn W, Popow C, Paky F (1996) Utilizing temporal data abstractions for data validation and therapy planning for artificially ventilated newborn infants. Artif Intell Med 8:543–576
27. Terenziani P, German E, Shahar Y (2008) The temporal aspects of clinical guidelines. In: Teije AT, Miksch S, Lucas P (eds) Computer-based medical guidelines and protocols: a primer and current trends
28. Aha D, Munoz-Avila H (2001) Introduction: interactive case-based reasoning. Appl Intell 14:78
29. Manago M, Althoff K-D, Auriol E, Traphoner R, Wess S, Conruyt N, Maurer F (1993) Induction and reasoning from cases. In: Proc European workshop on CBR. Springer, Berlin, pp 313–318
30. Aha D, Breslow L (1997) Refining conversational libraries. In: Proc international conference on case based reasoning. Springer, Berlin, pp 267–278
31. Aha D, Maney T, Breslow L (1998) Supporting dialogue inferencing in conversational case-based reasoning. In: Proc European workshop on case based reasoning. Springer, Berlin, pp 262–273
32. da Silva FJM, Sánchez Pérez JM, Gómez Pulido JA, Vega Rodríguez MA (2010) AlineaGA—a genetic algorithm with local search optimization for multiple sequence alignment. Appl Intell 32:164–172
33. Ross BJ (2000) Probabilistic pattern matching and the evolution of stochastic regular expressions. Appl Intell 13:285–300
34. Zhang W, Liu J, Niu Y (2009) Quantitative prediction of MHC-II peptide binding affinity using relevance vector machine. Appl Intell 31:180–187
35. Jin X, Xu A, Bie R (2008) Cancer classification from serial analysis of gene expression with event models. Appl Intell 29:35–46
36. González-Vélez H, Mier M, Julià-Sapé M, Arvanitis TN, García-Gómez JM (2009) HealthAgents: distributed multi-agent brain tumor diagnosis and prognosis. Appl Intell 30:191–202
37. Webb K, Whyte T (2006) Cell modeling with reusable agent-based formalisms. Appl Intell 24:169–181
38. Cho SB, Won H (2007) Cancer classification using ensemble of neural networks with multiple significant gene subsets. Appl Intell 26:243–250
39. Wu C, Chen H, Chen S (1997) Counter-propagation neural networks for molecular sequence classification: supervised LVQ and dynamic node allocation. Appl Intell 7:27–38
40. http://genome.lbl.gov/vista/index.shtml
41. Lassmann T, Sonnhammer ELL (2002) Quality assessment of multiple alignment programs. FEBS Lett 529:126–130

Stefania Montani received her PhD in Bioengineering from the University of Pavia in 2001. She is an associate professor in Computer Science at the University of Piemonte Orientale, in Alessandria, Italy. Her main research interests are in the areas of Case-Based Reasoning, Decision Support Systems, Business Process Management, Temporal Databases, Temporal Reasoning, and their applications in medicine and biology. She is the author of more than 140 papers in international journals and international refereed conferences in Artificial Intelligence and Medical Informatics.

Giorgio Leonardi earned a Laurea degree in computer science in 2004 and a PhD in Bioengineering and Bioinformatics in 2007. He is currently a post-doc researcher at the University of Piemonte Orientale "A. Avogadro" and at the University of Pavia. His research focuses on applying business process management/mining and Case-Based Reasoning on time-stamped data, service-flow management and knowledge abstraction and formalization through ontologies. The main application of these methodologies focuses on the medical domain.

Stefano Ghignone is a researcher at Plant Protection Institute (CNR). His research areas are genomics and functional genomics of symbiotic organisms (arbuscularmycorrhizal fungi and bacteria) and metagenomics/metagenetics of microbial soil communities. He is (co-)author of 20 peer-reviewed publications (h index 8).

Luisa Lanfranco is an associate professor at the Department of Life Science and Systems Biology (University of Torino) where she teaches Plant Molecular Biology. Her main scientific interests are genomics and functional genomics of arbuscularmycorrhizal fungi and plant responses to multiple microbes (symbiotic and pathogenic). She is (co-)author of 42 peer-reviewed publications (h index 17).