Flexible and interactive retrieval for supporting comparative genomics studies on endobacteria

S. Montani[1], G. Leonardi[1], S. Ghignone[2], L. Lanfranco[2]

[1] Dipartimento di Informatica, University of Piemonte Orientale, Alessandria, Italy
[2] Dipartimento di Biologia Vegetale, University of Turin, Italy

Abstract. Studying arbuscular mycorrhizal fungi and their symbiotic endobacteria has potentially strong impacts on the development of new biotechnology applications. Comparative genomics is a key technique for acquiring information about phylogenetic relationships and metabolic functions of such organisms. In this paper, we describe a case-based retrieval tool, which supports endobacteria genome sequence search and comparisons. While designing the tool, our main goal has been the one of granting for interpretability of the output results and understandability of the retrieval process. As a matter of fact, our system is supposed to provide support to biologists in a *flexible* and *interactive* way - and not to work in a black box fashion, providing complete (and unexplained) solutions to end users. To this end, we have implemented multi-level abstraction mechanisms and proper indexing techniques, for flexible query issuing, and interactive and efficient query answering. A case study taken from the application domain is used to illustrate the approach.

1 Introduction

Arbuscular mycorrhizal fungi (AMFs) are obligate symbionts which, to complete their life cycle, must enter in association with the root of land plants. Here, they become a crucial component of soil microbial communities, and exert positive impacts on plants health and productivity [1,2]. AMFs are thus a significant resource for sustainable agriculture; in addition, they could also be exploited as a still unknown resource to promote green (agriculture) and white (industrial) biotechnologies. AMFs are often in further symbiosis with uncultivable bacteria, living inside the AMF itself [3]. The resulting tripartite system (i.e. (i) endobacterium; (ii) AMF; (iii) plant roots) is a complex biological object, whose extensive study requires a comparative genomics approach, in order to answer fundamental questions concerning the biology, ecology and evolutionary history of the system and of its composing elements. As a matter of fact, comparative genomics represent a key instrument to discover or validate phylogenetic relationships, to give insights on genome evolution, and to infer metabolic functions of a given organism, which is particularly useful when biochemical and physiological data are not available and/or hard to obtain.

While bacterial endosymbionts in the animal kingdom are excellent models for investigating important biological events, such as organelle evolution, genome reduction, and transfer of genetic information among host lineages [4], examples of

endobacteria living in fungi are limited [5]. A key part of the study about the tripartite system mentioned above is therefore represented by the analysis of the genomic data of the endobacteria themselves. In particular, large-scale analysis and comparison of genomes belonging to phylogenetically related free-living bacteria can provide information about the events that led to genome down-sizing, and insights about the reason of the strict endosymbiotic life style of the bacteria at hand.

In this paper, we present a genome search and comparison tool, meant to support biologists in comparative genomics studies on such endobacteria. The tool relies on the *retrieval* step of the Case-Based Reasoning (CBR) methodology [6]. While designing the tool, our main goal has been the one of granting for interpretability of the output results and understandability of the retrieval process. As a matter of fact, our system is supposed to provide support to biologists in a *flexible* and *interactive* way - and not to work in a black box fashion, providing complete (and unexplained) solutions to end users. In our implementation, retrieval is made flexible by the possibility of expressing queries at different levels of abstraction detail, resorting to a technique similar to Temporal Abstractions [7,8]. 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. The tool is integrated into a modular architecture, composed by a database [9], in which massive genomic data are imported and stored, and by genomic comparison (synteny) and visualization tools, most of which are adaptations of open source software tools developed within the Generic Model Organism Database (GMOD) project [10]. Details of our work are provided in the next sections.

2 Flexible and interactive retrieval of similar genomes

The genome search and comparison tool we have designed, implements the *retrieval* step of the CBR [6] cycle. Note that *purely retrieval* systems are very valuable decision support tools [11], especially when automated adaptation strategies can be hardly identified, as in biology and medicine [12]; in our work we are following exactly this research line.

In our CBR 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 (see below for a detailed description of case representation). Completing the alignment task is therefore a prerequisite for representing cases in our library. In our approach we rely on BLAST (http://blast.ncbi.nlm.nih.gov/Blast.cgi) to deal with alignment. BLAST is in fact a state-of-the-art local alignment algorithm, specifically designed for bioinformatics applications. BLAST can take as an input any type of nucleotide sequence, i.e. a whole chromosome or plasmid, contigs or single genes, and properly aligns it to a database of strings belonging to (different) organisms of interest. From a typical BLAST output one can extract basic information (% of similarity and length of sequence alignment) that can be easily plotted as represented in figure 1, providing a piecewise constant function, which graphically represents the alignment itself.

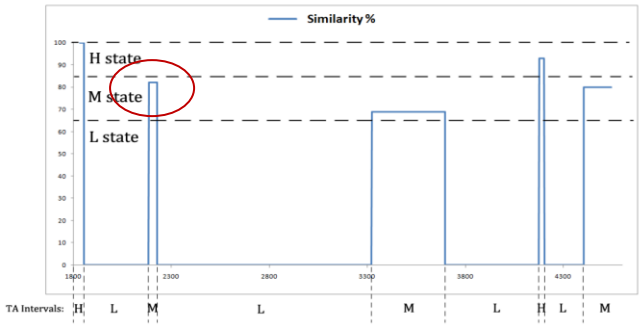

Fig. 1. A graphical visualization of sequence alignment (x-axis: nucleotide position of the alignment with respect to the reference string; y-axis: % of similarity). As an example, a substring with an 82% similarity, corresponding to a medium (M) qualitative value, is highlighted.

In our application domain, however, it is useful to convert the quantitative similarity values provided by BLAST to a set of qualitative levels (e.g. low, medium, 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 [7,8]. TA is an Artificial Intelligence methodology which allows 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 [8] allow to extract episodes associated with qualitative levels of the variable represented by the time series, where 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 percentages of similarity between the two strings themselves. An example is provided in figure 1, where e.g. a similarity of 82% is considered to be *medium* (M). More specifically, on the basis of domain knowledge we define a whole taxonomy of qualitative levels, where *low*, *medium* and *high* values can be further specialized (see figure 2). We also allow to abstract the information about the "localization" of the aligned substrings along the nucleotide sequences at different granularities (see figure 3), visualizing genome information at the level of 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

[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.

region may not. Thus, a high similarity at the region level might be difficultly identified at the level of single genes (see the case study below). Note that the taxonomy of granularities 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.

To summarize, in our framework ***case representation*** is obtained as follows. First, a pair of nucleotide sequences are optimally aligned by BLAST, which provides, for each aligned fragment, a percentage of similarity. Abstractions on such quantitative data are then calculated, and converted 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 figure 3). The resulting *string of symbols* (i.e. (very/moderately) low, medium, high) 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 conversion is actually the means by which we support ***flexible case retrieval***. In particular, a researcher can express queries as sequences of symbols, and may start with a generic, abstract requirement (e.g. look for a high similarity in the whole chromosome). All the cases in the case base exactly matching the query will be retrieved. Then, the user may progressively refine the query on the basis of the outputs s/he has collected (e.g. look for a very high similarity in the first gene), in order to reduce the retrieval set and better characterize the results. Refinement may take place in a way which cannot necessarily be foreseen a priori, interactively changing the level of abstraction/aggregation on demand.

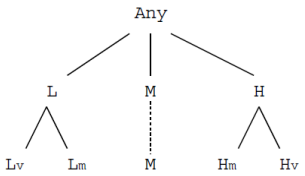

Fig. 2. An example taxonomy of state abstraction symbols; for instance, the high (H) symbol specializes into very high (Hv) and moderately high (Hm).

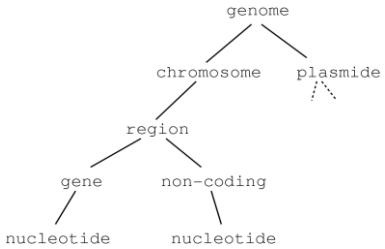

Fig. 3. A taxonomy of sequence granularities.

Obviously, since cases are stored at the ground level in both dimensions, in order to identify the cases that match a specific, possibly more abstract query, 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 must be provided. Moreover, a proper distance function must be defined for retrieval. The taxonomies, 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 - if any (e.g. the distance between L (*low*) and M (*medium*) is smaller than the distance between L (*low*) and H (*high*), see figure 2). Moreover, distance "preserves" ordering also in case *isa* relationships between symbols are involved (e.g. 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 [13].

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 taxonomies described above. 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 figure 2) 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 4. 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 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 as well, since, depending on the level at which the query is issued, according to the two taxonomies, one of the nodes can be more suited for providing a quick answer.

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

We will now illustrate query answering 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, murG*. The prominent feature of this cluster is that it is conserved with a high (H) 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 $H_vH_vL_vH_vH_vH_vH_vH_vH_vH_vH_vH_vH_vH_v$, looking for 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 figure 4), and then in the granularity dimension, providing the query H at the region level. This allows to enter the index in figure 4 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 e.g. 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 [13], and quickly described above) from the original query is greater than zero. 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 limited 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 figure 4), whose nodes can be exploded or iconified, facilitating index navigation (see figure 5). 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 (i.e. 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).

The interested reader may find additional technical details about our framework in [13]. Very encouraging experimental results have already been obtained by resorting to the same framework, in the field of haemodialysis [14].

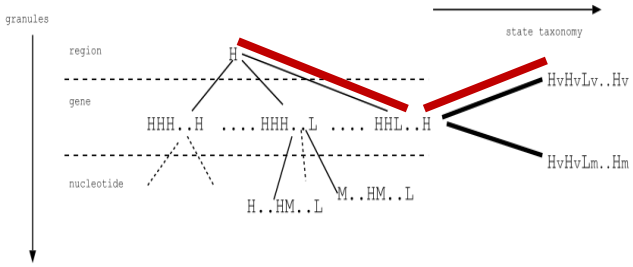

Fig. 4. An example multi-dimensional orthogonal index.

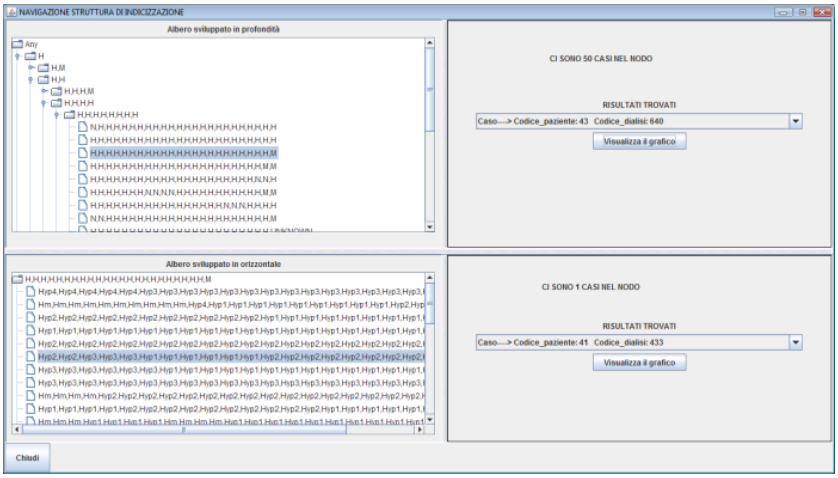

Fig. 5. A snapshot of index navigation as rendered by the system graphical interface.

3 Related work

In 1993, Aaronson [15] 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 [16] 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 [15] 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 [17] .

Except for the work in [15], however, we are not aware of CBR works in genomic comparison. Moreover, the work in [15] does not support any flexible and interactive case retrieval, as we are able to do by means of an abstraction mechanism.

As stated in section 2, our abstraction mechanisms resembles the one of TA [7,8]. In fact, the present work has been developed starting from our previous experience on TA-based time series retrieval [13,14], and properly adapting the characteristics of the framework described in [13,14] to the biological domain. Actually, such a framework was designed in a modular and domain-independent way. The adaptation to the biological domain has mainly consisted in resorting to *state* TA (instead of *trend* TA), which are more suited to deal with the abstraction of similarity levels. The adaptation has also required the acquisition of the specific domain knowledge, which has been the basis for a proper definition of the taxonomies and of the distance and *up* functions.

As regards TA, they have been extensively resorted to in the literature, especially in the medical field, from diabetes mellitus [18,19], to artificial ventilation of intensive care units patients [20] (see also the survey in [21]), but typically with the aim to solve a data interpretation task [7], and not as a flexible retrieval support facility. 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 was not to work in a black box fashion with respect to end-users: on the other hand, we wished to build an *interactive* system. It is worth noting that, in classical case retrieval 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. [22,23]). 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, with respect to the (abstract) query expressed so far, are listed, and further questions meant to reduce and specialize the retrieval set are asked by the system itself. Our framework thus loosely resembles CCBR. However, CCBR is characterized by some strong challenges, namely related to case authoring, and dialog inferencing. Both aspects are non-trivial, and should be solved by specific modules (based e.g. on machine learning [24] or model-based reasoning [25] techniques). Such an additional effort is not required in our framework.

5 Conclusions

In this paper, we have described the main features of a genome search and comparison 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, which has been designed by following software engineering principles, in order to enhance usability and user friendliness in the interaction with the system. Moreover, retrieval performances are optimized by resorting to multi-dimensional orthogonal index structures, allowing for a quick query answering.

In the next months, we will work at the system evaluation phase, by initializing the tool case base with a selection of genomes (in particular, from symbiotic microorganisms) available in the RefSeq NCBI database. A newly sequenced organism, in particular, will be made available. The tool will then support genomic comparison studies between the new genome and the ones of relative organisms already known to the scientific community, which is one of the key goals of the our application.

Acknowledgements

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. J. Marx. The roots of plant-microbe collaborations. *Science*, 304:234236, 2004.
2. P. Bonfante, A. Genre, . Mechanisms underlying beneficial plant-fungus interactions in mycorrhizal symbiosis, Nature Communications., 1:48, 2010.
3. I.A. Anca, E. Lumini, S. Ghignone, A. Salvioli, V. Bianciotto, P. Bonfante, The ftsZ gene of the endocellular bacterium 'Candidatus Glomeribacter gigasporarum' is preferentially expressed during the symbiotic phases of its host mycorrhizal fungus, Molecular plant-microbe interactions : MPMI;22(3):302-10, 2009
4. N.A. Moran, A.J. McCutcheon, and P. Nakabachi. Genomics and evolution of heritable bacterial symbionts. Annu. Rev. Genet., 42:165190, 2008.
5. E. Lumini, S. Ghignone, V. Bianciotto, and P. Bonfante. Endobacteria or bacterial endosymbionts? to be or not to be. New Phytol, 170:205208, 2006.
6. A. Aamodt and E. Plaza. Case-based reasoning: foundational issues, methodological variations and systems approaches. AI Communications, 7:3959, 1994.
7. Y. Shahar. A framework for knowledge-based temporal abstractions. Artificial Intelligence, 90:79133, 1997.
8. R. Bellazzi, C. Larizza, and A. Riva. Temporal abstractions for interpreting diabetic patients monitoring data. Intelligent Data Analysis, 2:97122, 1998.

9. C.J. Mungall, D.B. Emmert, The FlyBase Consortium. A chado case study: an ontology-based modular schema for representing genome-associated biological information. Bioinformatics, 23(13):i33746, July 2007.
10. B. Osborne and GMOD Community. GMOD, 2000.
11. I. Watson. Applying Case-Based Reasoning: techniques for enterprise systems. Morgan-Kaufmann, 1997.
12. S. Montani. Exploring new roles for case-based reasoning in heterogeneous AI systems for medical decision support. Applied Intelligence, 28:275285, 2008.
13. S. Montani, A. Bottrighi, G. Leonardi, L. Portinale, and P. Terenziani. Multi-level abstractions and multi-dimensional retrieval of cases with time series features. In L. McGinty and D. Wilson, editors, Proc. International Conference on Case-Based Reasoning (ICCBR) 2009, Lecture Notes in Artificial Intelligence 5650, pages 225 239. Springer-Verlag, Berlin, 2009.
14. A. Bottrighi, G. Leonardi, S. Montani, L. Portinale, P. Terenziani, Intelligent data interpretation and case base exploration through Temporal Abstractions, Proc. International Conference on Case-Based Reasoning (ICCBR) 2010, LNCS 6176, I. Bichindaritz, S. Montani eds., Springer, Berlin, pp. 36-50
15. J.S. Aaronson, H. Juergen, G.C. Overton, Knowledge discovery in GENBANK, Proc. International Conference on Intelligent Systems for molecular biology, L. Hunter, D. Searls, U. Shavlik eds., AAAI Press, pp. 3-11
16. I. Jurisica, J. Glasgow, "Applications of Case-Based Reasoning in molecular biology", AI Magazine, vol. 25(1), pp. 85-95, 2004.
17. I. Bichindaritz, A. Annest, Case based reasoning with Bayesian model averaging: an improved method for survival analysis on microarray data, Proc. International Conference on Case-Based Reasoning (ICCBR) 2010, LNCS 6176, I. Bichindaritz, S. Montani eds., Springer, Berlin, pp. 346-359
18. Y. Shahar and M. Musen, "Knowledge-based temporal abstraction in clinical domains," Artificial Intelligence in Medicine, vol. 8, pp. 267–298, 1996.
19. R. Bellazzi, C. Larizza, P. Magni, S. Montani, and M. Stefanelli, "Intelligent analysis of clinical time series: an application in the diabetes mellitus domain," Artificial Intelligence in Medicine, vol. 20, pp. 37–57, 2000.
20. S. Miksch, W. Horn, C. Popow, and F. Paky, "Utilizing temporal data abstractions for data validation and therapy planning for artificially ventilated newborn infants," Artificial Intelligence in Medicine, vol. 8, pp. 543–576, 1996.
21. P. Terenziani, E. German, and Y. Shahar, "The temporal aspects of clinical guidelines," in Computer-based Medical Guidelines and Protocols: A Primer and Current Trends, A. T. Teije, S. Miksch, and P. Lucas, Eds., 2008.
22. D. Aha and H. Munoz-Avila. Introduction: interactive case-based reasoning. Applied Intelligence, 14:78, 2001.
23. M. Manago, K.-D. Althoff, E. Auriol, R. Traphoner, S. Wess,N. Conruyt, and F. Maurer, "Induction and reasoning from cases," in Proc. European Workshop on CBR. Springer, 1993, pp. 313–318.
24. D. Aha and L. Breslow, "Refining conversational libraries," in Proc International Conference on Case Based Reasoning. Springer, 1997, pp. 267–278.
25. D. Aha, T. Maney, and L. Breslow, "Supporting dialogue inferencing in conversational case-based reasoning," in Proc European Workshop on Case Based Reasoning. Springer, 1998, pp. 262–273