Semantic Web Reasoning for Analyzing Gene Expression Profiles

Liviu Badea

AI Lab, National Institute for Research and Development in Informatics
8-10 Averescu Blvd., Bucharest, Romania
badea@ici.ro

Abstract. We argue that Semantic Web reasoning is an ideal tool for analyzing gene expression profiles and the resulting sets of differentially expressed genes produced by high-throughput microarray experiments, especially since this involves combining not only very large, but also semantically and structurally complex data and knowledge sources that are inherently distributed on the Web. In this paper, we describe an initial implementation of a full-fledged system for integrated reasoning about biological data and knowledge using Sematic Web reasoning technology and apply it to the analysis of a public *pancreatic cancer dataset* produced in the Pollack lab at Stanford.

1 Introduction and Motivation

The recent breakthroughs in genomics have allowed new rational approaches to the diagnosis and treatment of complex diseases such as cancer or type 2 diabetes. The role of *bioinformatics* in this domain has become essential, not just for managing the huge amounts of diverse data available, but also for extracting biological meaning out of heterogeneous data produced by different labs using widely different experimental techniques. Although the completion of the sequencing of the genomes of a large number of organisms (including the Human Genome) has identified the (more or less) complete lists of genes of these organisms, we only have a partial view of the complexity of the interactions among these genes.

Thus, determining the molecular-level details of complex diseases is a challenging issue. Traditional genetic methods are inapplicable since, typically, there is no single gene responsible for the disease. Rather, a complex interplay of pathways is usually involved, so that many *different* genetic (possibly somatic) defects[1] may affect the same pathway. Despite the large body of existing biological knowledge, even the pathways are only partially known and, even worse, may interact in very complex ways.

The study of complex diseases has been revolutionized by the advent of whole-genome measurements of gene expression using *microarrays*. These allow the determination of gene expression levels of virtually all genes of a given organism in a variety of different samples, for example coming from normal and diseased tissues.

However, the initial enthusiasm related to such microarray data has been tempered by the difficulty in their interpretation. It has become obvious that additional available

[1] Such as Single Nucleotide Polymorphisms (SNP), chromosomal translocations, chromosomal segment amplifications or deletions, etc.

J.J. Alferes et al. (Eds.): PPSWR 2006, LNCS 4187, pp. 78 – 89, 2006.

knowledge has to be somehow used in the data analysis process. However, the complexity of the types of knowledge involved renders any known data analysis algorithm inapplicable. Thus, we need to integrate at a deep semantic level the existing domain knowledge with the partial results from data analysis. *Semantic Web* technology, and especially the *reasoning* facilities that it will offer turn out to be indispensable in the biological domain at all levels:

- At the lower data access level, we are dealing with huge data- and knowledge bases that are virtually impossible to duplicate on a local server. A mediator-type architecture [16] would therefore be useful for integrating the various resources and for bridging their heterogeneity.
- At the level of data schemas, we frequently encounter in this domain very complex semi-structured data sources – accessing their contents at a semantic level requires precise machine-interpretable descriptions of the schemas.
- Finally, the data and knowledge refer to complex conceptual constructions, which require the use of common domain ontologies for bridging the *semantic* heterogeneities of the sources.

In this paper, we describe an initial attempt at developing a full-fledged system for integrated reasoning about biological data and knowledge using Semantic Web reasoning technology. The system is designed as an open system, able to quickly accommodate various data sources of virtually all types (semi-structured, textual, databases, etc.). At this time, we have a working system prototype that uses the state-of-the-art XML query language XQuery [9] for implementing the wrappers to the Web-based sources (either in XML or possibly non-well-formed HTML), the Flora2 [10] F-logic implementation for reasoning and a Tomcat-based implementation of the Web application server.

2 The Pancreatic Cancer Dataset

In the following we describe an application of the technology to the analysis of a public *pancreatic cancer dataset* produced in the Pollack lab at Stanford [1].

Despite the enormous recent progress in understanding cancer at a molecular level, the precise details are still elusive for many types of carcinomas. Pancreatic cancer is a particularly aggressive disease, with a very poor prognosis, requiring a more precise understanding of its molecular pathogenesis. The technological progress initiated by the introduction of gene expression microarrays about a decade ago has enabled large scale whole genome studies with the aim of identifying disease-specific genes. Although limited by the relatively low number of samples (due to the large costs of the technology), these gene expression studies have revealed a much more complex molecular-level picture than previously expected. Tens to a few hundreds genes were found to be differentially expressed in the samples analyzed, and their precise roles in the (signaling) pathways leading to cancer are only partially known. Even worse, it seems extremely difficult to discern between genetic abnormalities that play a causal role in oncogenesis and those that are merely side-effects. Obviously, the task of identifying new therapeutic targets depends essentially on being able to identify the causal details.

The results of published studies [1,2,3] have emphasized the complexity of the genetic abnormalities involved in pancreatic cancer. There seem to be few, if any, amplifications or deletions common to all patients thus leading to a more complex

picture of the disease in which perturbations of distinct components of certain key pathways are triggered in various different ways, while leading to similar phenotypes.

The fact that our knowledge of the various signaling pathways involved is only partial makes the task of identifying the precise details of oncogenesis even more difficult, requiring a combination of all the available data and knowledge.

More precisely, Bashyam et al. [1] have performed simultaneous *array Comparative Genomic Hybridization* and *microarray expression* measurements on a set of 23 human pancreatic cell lines (with two additional normal-normal reference array-CGH measurements) using cDNA microarrays containing 39632 human cDNAs (representing about 26000 named human genes). Array-CGH measurements involved co-hybridizing Cy5-labeled genomic DNA from each cell line along with Cy3-labeled sex-matched normal leukocyte DNA. Expression profiling was performed with reference RNA derived from 11 different human cell lines.

We retrieved the normalized intensity ratios from the Stanford Microarray Database [5] and used the CGH-Miner software [4] as described in [1] to identify DNA copy number gains and losses. Expression ratios were called significant if they $_{EXPR\_} = 0.5$.

Since for certain microarray spots expression ratios may be poorly defined (mainly due to low intensities in one of the two channels), we only retained genes whose expression ratios were well measured in at least 14 of the 23 samples. Unlike Bashyam et al. who performed mean centering of the (log-)expression ratios of the genes (to emphasize their relative levels among samples), we avoid mean-centering or variance normalization of the ratios since we are interested in identifying systematically over/under-expressed genes, the expression level being important for this purpose. Finally, we constructed two lists of "common" up- and respectively down-regulated genes *Common* and *Common* , which we use in the following.

3 The Data Sources

The architecture of the application is presented in Figure 2 in the Appendix. The application uses various data and knowledge sources, ranging from semi-structured data to databases of literature-based paper abstracts.

We initially integrated the following sources:

NCBI/Gene. The e-utilities [11] interface to the NCBI Gene database [12] returns gene-centred information in XML format. We extracted using an XQuery wrapper gene symbols, names, descriptions, protein domains (originating from Pfam or CCD), and literature references. We also extracted the Gene Ontology (GO) [13] annotations of the genes, as well as the pathways[2] and interactions[3] in which these are known to be involved.

TRED. The Transcriptional Regulatory Element Database TRED [8] contains knowledge about transcription factor binding sites in gene promoters. Such information is essential for determining potentially co-expressed genes and for linking them to signaling pathways.

[2] Originating from KEGG or Reactome.
[3] Taken e.g. from BIND or HPRD.

Biocarta [7] is a pathway repository containing mostly graphical representations of pathways contributed by an open community of researchers. We have developed an XQuery wrapper that currently extracts the lists of genes involved in the various pathways.

Pubmed. Literature references to genes and their interactions extracted from Pubmed abstracts [14] will also be integrated into the system.

The above sources contain complementary information about the genes, their interactions and pathways, neither of which can be exploited to their full potential in isolation. For example, the GO annotations of genes can be used to extract the main functional roles of the genes involved in the disease under study. Many such genes are receptors or their ligands, intra-cellular signal transducers, transcription factors, etc. And although many of these genes are known to be involved in cancer (as oncogenes or tumor suppressors), the GO annotations will not allow us to determine their interactions and pathway membership. These can only be extracted from explicit interaction or pathway data-sources, such as TRED, BIND, Biocarta, etc.

4 A Unified Model of the Data Sources

In order to be able to jointly query the data sources, a unified model is required. We used the prototype system described in [17] to implement a mediator over the above-mentioned data sources. The system uses *F-logic* [23] for describing the content of information sources as well as the domain ontology for several important reasons.

First, although the distinctive feature of the Semantic Web is reasoning, the various related W3C standards are not easy to use by a reasoner, especially due to their heterogeneity (XML, RDF, RuleML, etc.). A *uniform* internal level, optimized for efficiency is required for supporting inference and reasoning. The architecture of our system therefore separates a so-called *"public" level* from the *internal level.* The public level refers to the data, knowledge and models exchanged on the Web and between applications and conforms to the current and emerging Web standards such as XML, RDF(S), RuleML, etc. F-logic is used at the "internal" level.

Second, the tabling mechanism of Flora2 [4] is essentially equivalent to the Magic Sets method [24] for bottom-up evaluation in database query engines, which, combined with top-down evaluation, can take advantage of the highly optimized compilation techniques developed for Prolog, resulting in a very efficient deductive engine.

Moreover, F-logic combines the logical features of Prolog with the frame-oriented features of object-oriented languages, while offering a more powerful query language (allowing e.g. aggregation and meta-level reasoning about the schema). Last but not least, F-logic is widely used in the Semantic Web community [18,19,20]. However, we also consider the possibility of using Xcerpt [21] at this level.

4.1 Mapping Rules

Since the sources are heterogeneous, we use so-called "*mapping rules*" to describe their content in terms of a common representation or ontology. For example, we can retrieve direct interactions either from the gene-centred NCBI Gene database, or from TRED:

[4] Flora2 is the F-logic implementation we use.

```
di(I):direct_interaction[gene->G1, other_gene->G2, int_type->IntType, source->'ncbi_gene',
                                    description->Desc, pubmed->PM] :–
    query_source('ncbi_gene_interactions', 'bashyam')@query,
    I:interaction[gene->G1, other_gene->G2, description->Desc,
                  pubs->PM]@'ncbi_gene_interactions',
    if (str_sub('promoter',Desc,_)@prolog(string))
    then IntType = 'p-d'
    else IntType = 'p-p'.

di(I):direct_interaction[gene->G1, other_gene->G2, int_type->IntType, source->'tred'] :–
    query_source('tred', 'bashyam')@query,
    I:interaction[tf->G1, gene->G2]@'tred',
    IntType = 'p-d'.
```

The common representation refers to direct interactions by the direct_interaction Flora2 object. We distinguish between two types of interactions:

- protein-to-DNA (‘*p-d*’), which refers to transcription regulatory influences between a protein and a target gene, and
- protein-to-protein (‘*p-p*’), which comprises all other types of interactions.

The distinction is important since the gene expression data analyzed reveals only changes in expression levels. Thus, while the protein-to-DNA interactions could in principle be checked against the expression data, the protein-to-protein interactions are complementary to the expression data[5] and could reveal the cellular functions of the associated proteins.

While certain types of knowledge are more or less explicit in the sources (for example, the interaction type is ‘*p-d*’ if the description of the interaction contains the substring ‘*promoter*’), in other cases we may have to describe implicit knowledge about sources (i.e. knowledge that applies to the source but cannot be retrieved from it – for example, the TRED database contains only interactions of type ‘*p-d*’, but this is nowhere explicitly recorded in the data).

4.2 Model Rules

Although in principle the wrappers and the mapping rules are sufficient for being able to formulate and answer any query to the sources, it is normally convenient to construct a more complex model, that is as close as possible to the conceptual model of the users (molecular biologists/geneticists in our case). This is achieved using so called “*model rules*” which refer to the common representation extracted by the mapping rules to define the conceptual view (model) of the problem.

For example, we may want to query the system about “*functional*” interactions (which are not necessarily *direct* interactions). More precisely, a functional interaction between two genes can be either due to a direct interaction, or to the membership in the same pathway, or to their co-reference in some literature abstract from Pubmed:

```
pi(I1,I2):pathway_interaction[gene->G1, other_gene->G2, int_type->IntType,
                                source->[Src1,Src2], pathway->P, role(G1)->R1, role(G2)->R2] :
    I1:pathway[name->P, gene->G1, gene_description->GN1, role(G1)->R1, source->Src1],
    I2:pathway[name->P, gene->G2, gene_description->GN2, role(G2)->R2, source->Src2],
    interaction_type(R1,R2,IntType).
```

[5] i.e. cannot be derived from it.

```
interaction_type(target_gene, target_gene, coexpression) :  !.
interaction_type(target_gene, Role2, transcriptional) :  Role2 \= target_gene, !.
interaction_type(Role1, target_gene, transcriptional) :  Role1 \= target_gene, !.
interaction_type(Role1, Role2, same_pathway) :  Role1 \= target_gene, Role2 \= target_gene, !.

fi(I):functional_interaction[gene->G1, other_gene->G2, int_type->IntType, source->Src] :
      I:direct_interaction[gene->G1, other_gene->G2, int_type->IntType, source->Src]
      ; I:pathway_interaction[gene->G1, other_gene->G2, int_type->IntType, source->Src]
      ; I:literature_interaction[gene->G1, other_gene->G2, int_type->IntType, source->Src].
```

We may also define classes of genes based on their GO annotations. For example, the following rules extract receptors, ligands and respectively transcription regulators:

```
r(I):gene_role[gene->G, category->C, role->receptor, source->Src] :
      I:gene_category[gene->G, category->C, source->Src],
      str_sub('receptor',C,_)@prolog(string),
      str_sub('activity',C,_)@prolog(string).

r(I):gene_role[gene->G, category->C, role->ligand, source->Src] :
      I:gene_category[gene->G, category->C, source->Src],
      str_sub('receptor',C,_)@prolog(string),
      ( str_sub('binding',C,_)@prolog(string) ;
        str_sub('ligand',C,_)@prolog(string) ).

r(I):gene_role[gene->G, category->C, role->transcription_regulator, source->Src] :
      I:gene_category[gene->G, category->C, source->Src],
      ( str_sub('DNA binding',C,_)@prolog(string) ;
        str_sub('transcription',C,_)@prolog(string) ).
```

Such classes of genes can be used to "fill in" *templates* of signaling chains, such as ligand → receptor → signal transducer →…→ transcription factor, which could in principle be reconstructed using knowledge about interactions:

```
generic_signaling_chain_interaction(ligand, receptor, 'p-p').
generic_signaling_chain_interaction(receptor, signal_transducer, 'p-p').
generic_signaling_chain_interaction(signal_transducer,  signal_transducer, 'p-p').
generic_signaling_chain_interaction(signal_transducer, transcription_factor, 'p-p').
generic_signaling_chain_interaction(transcription_factor, target_gene, 'p-d').
generic_signaling_chain_interaction(modulator, receptor, 'p-p').
generic_signaling_chain_interaction(modulator, signal_transducer, 'p-p').
generic_signaling_chain_interaction(modulator, transcription_factor, 'p-p').

signaling_chain(sig_chain(G), G, Role) :

      Role = receptor,
      _:gene_role[gene->G, role->Role].
signaling_chain(S, G2, Role2) :

      signaling_chain(S, G1, Role1),
      generic_signaling_chain_interaction(Role1, Role2, IntType),
      _:direct_interaction[gene->G1, other_gene->G2, int_type->IntType],
      _:gene_role[gene->G2, role->Role2].
```

Note that the signaling chains are initialized with receptors, since these are the starting points of signaling cascades and are typically affected in most cancer samples (including our pancreatic cancer dataset).

In our cancer dataset analysis application, the transcription factors play an important role, since their gene targets' co-expression can reveal the groups of genes that are differentially co-regulated in the disease:

```
tf_binding(G1, G2, IntType) :
    _:gene_role[gene->G1, category->C1, role->transcription_regulator],
     _:direct_interaction[gene->G1, other_gene->G2, int_type->IntType, source->Src],
    _:gene_list[gene->G2, list->common].
```

Figure 1 below shows the graph generated by the system in response to the following query (Cytoscape [22] is used for visualization):

```
?- show_graph(${tf_binding(TF,G,IntType)}, [TF,G,IntType]).
```

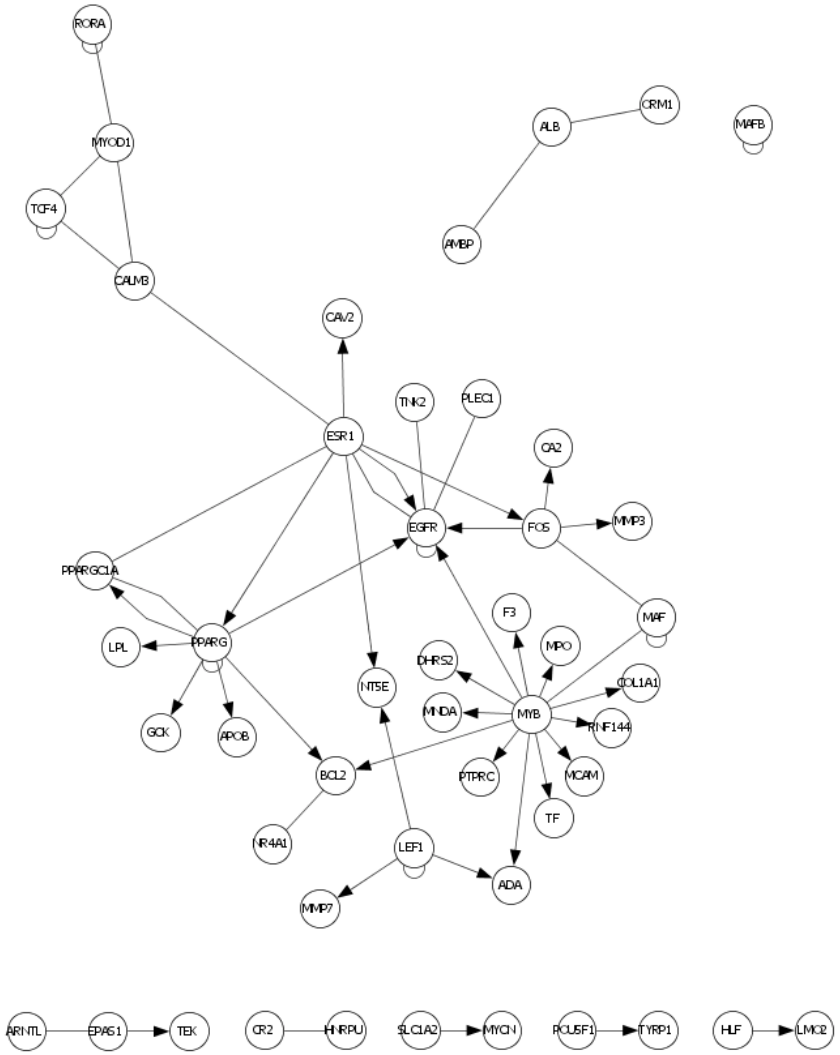

Fig. 1. Transcription regulatory relationships among "common" genes in the Bashyam et al. pancreatic cancer dataset (arrows: 'p-d', undirected edges: 'p-p' interactions)

5 Conclusions and Future Work

Our initial experiments confirmed the feasibility of our approach and lead to a number of interesting observations. Although all processing was performed in-memory, the system was able to deal with the complete data-sources mentioned above for the selection of "common" genes (359 genes):

- NCBI Gene interactions: 2239
- TRED interactions: 10717
- Biocarta gene to pathway membership relations: 5493
- NCBI gene to pathway membership relations: 622
- Other pathway membership relations: 5095
- GO annotations: 2394
- Protein domains: 614.

From a certain perspective, the approach is a combination of *remote-source mediation* and *data-warehousing*. As in a mediation approach, only the *relevant* entries of remote data sources are retrieved, but these are stored in a local warehouse by the wrappers (in XML format) to avoid repetitive remote accesses over the Web.

Such exploratory queries involving large datasets and combinatorial reasoning typically have slow response times (typically seconds to minutes if the relevant sources have been accessed previously and are therefore in the local warehouse; if not, response times depend on the size of the data to be transferred from remote sources and on the connection speed). However, as far as we know, other existing approaches are either slower[6] or cannot deal with such datasets at all.

Such exploratory queries involving large datasets and combinatorial reasoning typically have slow response times (typically seconds to minutes if the relevant sources have been accessed previously and are therefore in the local warehouse; if not, response times depend on the size of the data to be transferred from remote sources and on the connection speed). However, as far as we know, other existing approaches are either slower or cannot deal with such datasets at all.

Since reasoning in general is based on *combining* knowledge, Semantic Web reasoning will have to deal with combining knowledge *distributed* on the Web. The distributed nature of relevant knowledge in turn places significant limitations on the reasoners, due to the limited data transfer speeds of the current Web. Thus, it appears that future Semantic Web reasoning systems will be placed between two extremes, depending on the scope of the knowledge relevant to a query. At one extreme, there will be general, Google-like systems that will use local warehouses of the entire Web for answering semantic queries. At the other extreme, Web browsers will be enhanced with (semantic) reasoning capabilities, but the reasoning will be performed on a single Web page only. Our approach comes somehow in between the two extremes: the relevant and frequently used sources and Web pages are stored in a local warehouse allowing more sophisticated queries than in the "browser only" setting.

[6] In the case of systems based on plain Prolog (with no tabling or other similar optimizations).

We have also tried to implement fragments of the above scenario using XQuery not just for the wrappers, but also for the integrated model. (In our experiments, we have used the qizxopen [9] implementation of XQuery. The general idea consisted in implementing the reasoning rules as XQuery functions.) Although the efficiency and memory consumption are comparable to those of our F-logic-based system, using a procedural query language like XQuery posed significant problems. For example, the following XQuery function retrieves the transcription regulatory interactions involving common genes:

```
declare function local:select-NCBI_Gene-tranreg_interactions_common($NCBI_Gene_common
as node(), $common_genes as xs:string  *) as node() *
{
<RESULTS>
{
   for $int in $NCBI_Gene_common//interaction,
       $g1 in $common_genes[. = string($int/gene)],
       $g2 in $common_genes[. = string($int/other_gene)]
   let $g := $int/../..
   where contains(lower-case(string($g/Gene_Ontology/GO_category/GO_annot/GO)),
   "transcription")
   return
   <transcription_regulator_interaction_common>{$int/*}</transcription_regulator_interaction_com
   mon>
}
</RESULTS>
}
```

Note the rather complex way of performing simple operations such as joins. But even if we ignore such syntactic complications, we would have to write a separate XQuery function *for each possible instantiation pattern* of a given rule head, leading to a cumbersome and hard to modify program (a modification of a rule would require synchronized modifications in all associated XQuery functions).

Finally, there are certain technical issues whose improvement would lead to a significantly better Semantic Web reasoning system:

- Query planning
- Streaming
- Source capabilities
- Support for (semi-)automated development of wrappers.

In the case of large data sources, as in the biological domain (giga- to terrabytes), it is obviously impossible to retrieve the *entire* content of such sources before starting reasoning. Also, if additional knowledge is available about the sources, some source accesses may be avoided altogether. Therefore, dealing with information sources requires a certain form of *query planning*, i.e. the ability of constructing and reasoning about alternative sequences of source accesses (plans) before actually querying these sources. Also, *streaming the query responses* may allow starting processing before the entire response is retrieved.

Since queries can involve *several* different information sources, they will have to be to be *split* into sub-queries that can be treated by the separate information

sources. Since each information source may have its own (Web accessible) interface, we need to explicitly represent the *capabilities* of these interfaces. As opposed to traditional database query languages, such Web sources provide only limited query capabilities. For example, a specific Web interface may allow only certain types of selections and may also require certain parameters to be inputs (i.e. known at query time). These source capabilities would have to be taken into account during query planning.

From the biological point of view, the system has proved to be very useful for creating a global "picture" of the interactions among the genes differentially expressed in pancreatic cancer. The large number (359) of these genes [7] would have made the task extremely difficult, if not impossible for a human exploration of the data sources. For example, note the involvement of: [8]

- the Epidermal Growth Factor Receptor EGFR, known to be involved in many cancers
- BCL2, a gene involved in the apoptotic response of cells (note that the down-regulation of BCL2 in pancreatic cancer is quite unusual for an anti-apoptotic gene, since it is normally over-expressed in other tumor types [15])
- the transcription factors FOS, MYB, LEF1
- the metalloproteinases MMP3, and MMP7 (involved in tissue remodeling, invasion, tumor progression, metastasis and tumor initiation – in the case of MMP3)
- the nuclear receptor PPARG, a regulator of differentiation known to be involved in cancer and PPARGC1A, its coactivator.

The biological interpretation of the results is outside the scope of this paper and will be discussed elsewhere in a specialized paper.

Acknowledgements. I am grateful to Doina Tilivea for her contribution in implementing the F-logic system [17] and to Anca Hotaran for contributing to the development of the XQuery wrappers. This research has been partially funded by the European Commission within the 6th Framework Programme project REWERSE (506779, http://rewerse.net). I am deeply grateful to the REWERSE members for interesting discussions during the conference and for supporting this research.

References

1. Bashyam MD et al.Array-based comparative genomic hybridization identifies localized DNA amplifications and homozygous deletions in pancreatic cancer.Neoplasia. 2005 Jun;7(6):556-62
2. Heidenblad M et al. Genome-wide array-based comparative genomic hybridization reveals multiple amplification targets and novel homozygous deletions in pancreatic carcinoma cell lines. Cancer Res. 2004 64(9):3052-9.

[7] Amounting to 64261 potential interactions.
[8] See Figure 1.

3. Heidenblad M et al. Microarray analyses reveal strong influence of DNA copy number alterations on the transcriptional patterns in pancreatic cancer: implications for the interpretation of genomic amplifications. Oncogene. 2005 Mar 3;24(10): 1794-801.
4. Wang P, Kim Y, Pollack J, Narasimhan B, Tibshirani R. A method for calling gains and losses in array CGH data. Biostatistics. 2005 Jan;6(1):45-58.
5. Sherlock G. et al. The Stanford Microarray Database. Nucleic Acids Research, 29:152--155, 2001. http://genome-www5.stanford.edu
6. Bhattacharjee et al. Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses. Proc. Natl. Acad. Sci. USA. 2001 Nov. 20;98(24):13790-5.
7. Biocarta. www.biocarta.com
8. Fang Zhao, Zhenyu Xuan, Lihua Liu, Michael Q. Zhang. TRED: a Transcriptional Regulatory Element Database and a platform for in silico gene regulation studies. Nucleic Acids Res. 2005 January 1; 33(Database Issue): D103–D107.
9. Qizxopen. http://www.xfra.net/qizxopen/
10. Yang G., Kifer M., Zhao C. FLORA-2: A Rule-Based Knowledge Representation and Inference Infrastructure for the Semantic Web. In Second International Conference on Ontologies, Databases and Applications of Semantics (ODBASE), Catania, Sicily, Italy, November 2003. http://flora.sourceforge.net/
11. NCBI e-utilities. http://eutils.ncbi.nlm.nih.gov/entrez/query/static/eutils_help.html
12. NCBI Gene. http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=gene
13. Ashburner M. et al. Gene ontology: tool for the unification of biology. Nat. Genet. 2000 May;25(1):25-9. http://www.geneontology.org
14. Pubmed. http://www.ncbi.nih.gov/entrez/query.fcgi?db=PubMed
15. Westphal S, Kalthoff H.Apoptosis: targets in pancreatic cancer.Mol Cancer. 2003 Jan 7;2:6. Review.
16. Wiederhold G. Mediators in the architecture of future information systems, IEEE Comp. 25(3) 1992, 38-49.
17. Liviu Badea, Doina Tilivea, Anca Hotaran. Semantic Web Reasoning for Ontology-Based Integration of Resources. Proc. PPSWR 2004, pp. 61-75, Springer Verlag.
18. Decker S., Sintek M. 'Triple - an RDF query, inference, and transformation language', in Proc. of the 2002 International Semantic Web Conference (ISWC-2002).
19. Fensel D., Angele J., Decker S., Erdmann M., Schnurr H.P., Staab S., Studer R., Witt A., On2broker: Semantic-based Access to Information Sources at the WWW, Proceedings of WebNet, 1999, pp. 366-371.
20. Ludascher B., Himmeroder R., Lausen G., May W., Schlepphorst C. Managing Semistructured Data with FLORID: A Deductive Object-oriented Perspective. Information Systems, 23(8):589-613, 1998.
21. Berger S., Bry F., Schaffert S., Wieser C. Xcerpt and visXcerpt: From Pattern-Based to Visual Querying of XML and Semistructured Data. Proceedings VLDB03, Berlin, September 2003, http://www.xcerpt.org/.
22. Cytoscape. http://www.cytoscape.org
23. Kifer M., Lausen G., Wu J. Logical Foundations of Object-Oriented and Frame-Based Languages. Journal of the ACM, 42:741-843, 1995.
24. Bancilhon F., Maier D., Sagiv Y. and Ullman J. Magic sets and other strange ways to implement logic programs. In Proceedings PODS (1986) 1-15.

Appendix

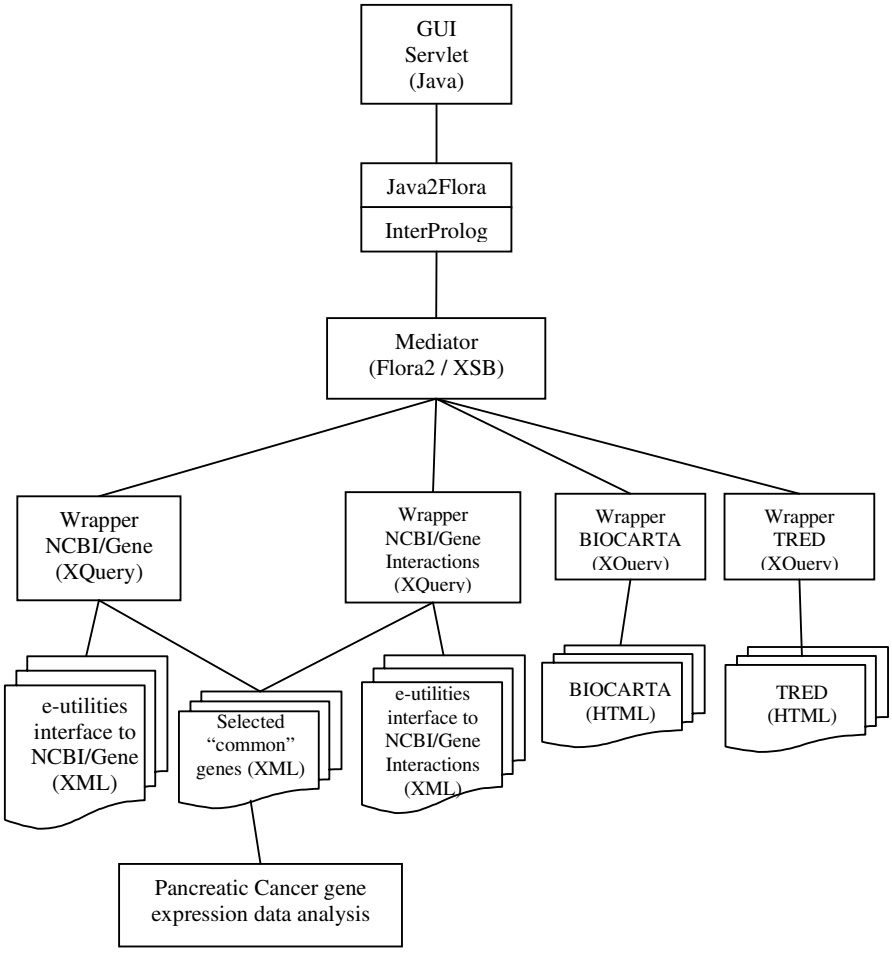

Fig. 2. The architecture of the pancreatic cancer dataset analysis application