

Analyzing Gene Expression Profiles with Semantic Web Reasoning

(Extended Abstract)

Liviu Badea¹, Doina Tilivea¹, Anca Hotaran¹

¹AI Lab, National Institute for Research and Development in Informatics
8-10 Avereșcu 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 Semantic Web reasoning technology and apply it to the analysis of a public *pancreatic cancer dataset* produced in the Pollack lab at Stanford.

1 Introduction

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.

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 knowledge has to be somehow used in the data analysis process. However, the complexity of the types of knowledge involved render 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 [15] 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 the following 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 are using the state-of-the-art XML query language XQuery [8] for implementing the wrappers to the Web-based sources (either in XML or possibly non-well-formed HTML), the Flora2 [9] 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]. 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 [4] and used the CGH-Miner software as described in [1] to identify DNA copy number gains and losses. Expression ratios were called significant if they either exceeded the threshold $\theta_{EXPR+} = 2$, or were below $\theta_{EXPR-} = 0.5$.

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.

We constructed the following two lists of “common” up- and respectively down-regulated genes: *Common+* and *Common-*

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 [10] interface to the NCBI Gene database [11] returns gene-centred information in XML format. We extracted using an XQuery wrapper gene symbols, names, descriptions, domains (originating from Pfam or CCD), and literature references. We also extracted the Gene Ontology (GO) [12] annotations of the genes, as well as the pathways¹ and interactions² in which these are known to be involved.

TRED. The Transcriptional Regulatory Element Database TRED [7] 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.

Biocarta [6] is 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 [13] 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.

¹ Originating from KEGG or Reactome.

² Taken e.g. from BIND or HPRD.

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 [16] to implement a mediator over the above-mentioned data sources. The system uses *F-logic* [9] for describing the content of information sources as well as the domain ontology. However, we also consider the possibility of using Xcerpt [17] 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:

```
di(l):direct_interaction[gene->G1, other_gene->G2, int_type->IntType, source->'ncbi_gene',
description->Desc, pubmed->PM] :-
  query_source('ncbi_gene_interactions', 'bashyam')@query,
  !: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(l):direct_interaction[gene->G1, other_gene->G2, int_type->IntType, source->'tred'] :-
  query_source('tred', 'bashyam')@query,
  !: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 data³ 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.

³ i.e. cannot be derived from it.

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(l1,l2):pathway_interaction[gene->G1, other_gene->G2, int_type->IntType,
    source->[Src1,Src2], pathway->P, role(G1)->R1, role(G2)->R2] :-
    l1:pathway[name->P, gene->G1, gene_description->GN1, role(G1)->R1, source->Src1],
    l2:pathway[name->P, gene->G2, gene_description->GN2, role(G2)->R2, source->Src2],
    interaction_type(R1,R2,IntType).

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(l):functional_interaction[gene->G1, other_gene->G2, int_type->IntType, source->Src] :-
    l:direct_interaction[gene->G1, other_gene->G2, int_type->IntType, source->Src]
    ; l:pathway_interaction[gene->G1, other_gene->G2, int_type->IntType, source->Src]
    ; l: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(l):gene_role[gene->G, category->C, role->receptor, source->Src] :-
    l:gene_category[gene->G, category->C, source->Src],
    str_sub('receptor',C,_)@prolog(string),
    str_sub('activity',C,_)@prolog(string).

r(l):gene_role[gene->G, category->C, role->ligand, source->Src] :-
    l: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(l):gene_role[gene->G, category->C, role->transcription_regulator, source->Src] :-
    l: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, source->Src],
    _: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 [18] is used for visualization):

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

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
- 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 slower⁴ or cannot deal with such datasets at all.

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.

⁴ In the case of systems based on plain Prolog (with no tabling or other similar optimizations).

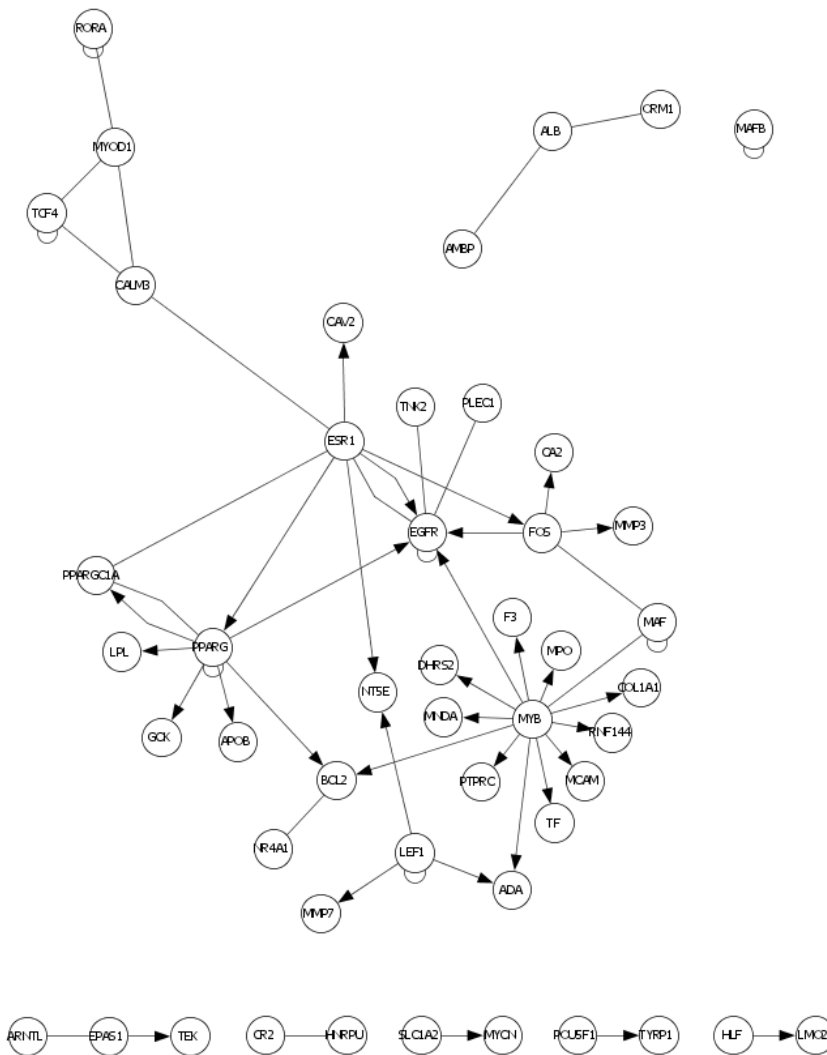


Figure 1. Transcription regulatory relationships among “common” genes in the Bashyam et al. pancreatic cancer dataset (arrows: ‘p-d’, undirected edges: ‘p-p’ interactions)

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 would have made the task extremely difficult, if not impossible for a human exploration of the data sources. For example, note the involvement of:

- the Epidermal Growth Factor Receptor EGFR, known to be involved in any 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 [14])
- 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.

6 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. Westphal S, Kalthoff H. Apoptosis: targets in pancreatic cancer. *Mol Cancer*. 2003 Jan 7;2:6. Review.
3. Lipson D, et al. Joint Analysis of DNA Copy Numbers and Gene Expression Levels Proceedings of Algorithms in Bioinformatics: 4th International Workshop, WABI 2004, Bergen, Norway, September 17-21, 2004, Lecture Notes in Computer Science (LNCS), Vol. 3240/2004, p.135, Springer 2004.
4. The Stanford Microarray Database. <http://genome-www5.stanford.edu>
5. 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.
6. Biocarta. www.biocarta.com
7. TRED. <http://rulai.cshl.edu/cgi-bin/TRED/tred.cgi?process=home>
8. Qizxopen. <http://www.xfra.net/qizxopen/>
9. Flora2. <http://flora.sourceforge.net/>
10. NCBI e-utilities. http://eutils.ncbi.nlm.nih.gov/entrez/query/static/eutils_help.html
11. NCBI Gene. <http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=gene>
12. Gene Ontology. <http://www.geneontology.org/>
13. Pubmed. <http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=PubMed>
14. Westphal S, Kalthoff H. Apoptosis: targets in pancreatic cancer. *Mol Cancer*. 2003 Jan 7;2:6. Review.
15. Wiederhold G. Mediators in the architecture of future information systems, *IEEE Comp*. 25(3) 1992, 38-49.
16. Liviu Badea, Doina Tilivea, Anca Hotaran. Semantic Web Reasoning for Ontology-Based Integration of Resources. *Proc. PPSWR 2004*, pp. 61-75, Springer Verlag.
17. 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/>.
18. Cytoscape. <http://www.cytoscape.org>

Appendix.

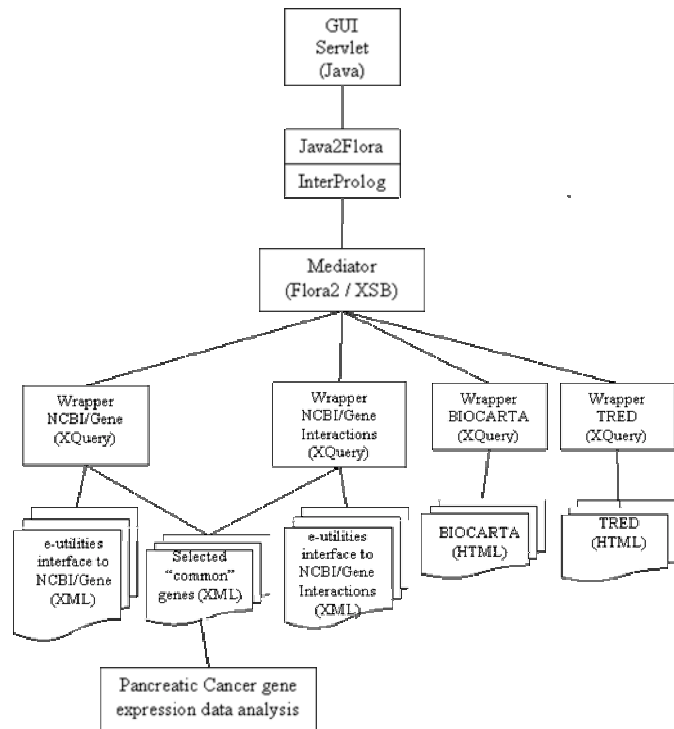


Figure 2. The architecture of the pancreatic cancer dataset analysis application