Web-Enabled Knowledge-Based Analysis of Genetic Data

Peter Juvan[1], Blaž Zupan[1,3], Janez Demšar[1], Ivan Bratko[1,2],
John A. Halter[4], Adam Kuspa[5,6], and Gad Shaulsky[6]

[1] Faculty of Computer and Information Science, University of Ljubljana, Slovenia
[2] Jožef Stefan Institute, Ljubljana, Slovenia
[3] Office of Information Technology and Department of Family and Community Medicine
[4] Division of Neuroscience
[5] Department of Biochemistry and Molecular Biology
[6] Department of Molecular and Human Genetics
Baylor College of Medicine, Houston, TX, USA

Abstract. We present a web-based implementation of GenePath, an intelligent assistant tool for data analysis in functional genomics. GenePath considers mutant data and uses expert-defined patterns to find gene-to-gene or gene-to-outcome relations. It presents the results of analysis as genetic networks, wherein a set of genes has various influence on one another and on a biological outcome. In the paper, we particularly focus on its web-based interface and explanation mechanisms.

1 Introduction

Genetic research is one of the most effective approaches to the elucidation of biological processes. State-of-the-art technology has enabled the genomic sequencing of several simple organisms as well as complex organisms including human. While obtaining a genome sequence is an important milestone, it marks only the beginning of the effort needed to understand and use this knowledge. The hardest step awaits us in the field of functional genomics [6], which is concerned with the "development and application of global (genome-wide or system-wide) experimental approaches to assess gene function" [3].

Analysis of mutant data, a core task in functional genomics, has so far been performed manually. With advancing technology that enables higher rates of experimentation and data gathering, new computer-based approaches are required to support data analysis in functional genomics. We have developed a program called GenePath that provides intelligent assistance for analysis of genetic data. In discovery of gene functions, GenePath mimics expert geneticist by using the reasoning patterns geneticists would else employ manually. GenePath capitalizes on computer's processing power to systematically examine the data, thus automatizing potentially tiresome and error-prone process.

GenePath is implemented in Prolog [1] and for its reasoning uses elements of selected approaches from artificial intelligence. While its environment in Prolog was

sufficient for prototyping and preliminary testing, participating geneticists strongly indicated that a more standard interface is needed. This pushed us towards the development of web-based interface (`http://magix.fri.uni-lj.si/genepath`), where Prolog-based core is seamlessly integrated within server-based application.

Basic elements of GenePath's core have been described in [8]. We here present the basics of its reasoning process, and then focus on its web-based implementation and methods that aim to explain its findings.

2 GenePath's Reasoning System

An input to GenePath is a set of genetic experiments, where each describes which genes were mutated and how and gives a qualitatively observed outcome of experiment. For example, consider a set of experiments on *Dictyostelium discoideum*, a soil ameba that is the subject of study in the laboratories of the authors from Baylor College of Medicine. *Dictyostelium* is particularly interesting for its social behavior and development cycle from single independent cells to a multicellular slug like form [7, 5, 4]. A sub process of *Dictyostelium*'s development is cell aggregation, which has been observed in experiments from Table 1.

Table 1. Genetic experiments: aggregation of *Dictyostelium*

Exp ID	Genotype	Aggregation
1	wild-type	+
2	yakA:: -	-
3	pufA:: -	++
4	yakA:: -, pufA:: -	++

Notice that in the first experiment from Table 1 no mutations were made and under normal conditions *Dictyostelium* aggregates. Under the same conditions, but with loss-of-function mutation of gene yakA (experiment 2), the aggregation does not take place. Additionally mutating pufA, aggregation is restored and actually takes place faster compared to wild-type *Dictyostelium* (experiment 4).

When geneticists analyze such data, they most often use a set of informal, unwritten but intuitive rules to derive relations between genes and outcome. For instance, an example of a simple rule is "*IF mutation of gene A changes the outcome P (compared to the wild type) THEN gene A is influencing the outcome P*". Using this rule, it can be concluded from Table 1 that both genes yakA and pufA influence the aggregation. An example of a more complex rule is "*IF mutation of gene A changes the outcome P (compared to the wild type) and adding the mutation of gene B reverses the outcome P THEN gene B acts after gene A in a path for the outcome P*". Using this rule and experiments 2 and 4 from Table 1, it can be concluded that pufA acts after yakA in the path for aggregation.

GenePath currently includes about 10 such rules, which are also referred to as patterns. It uses them to abduce the following gene-to-gene or gene-to-outcome relations:

1. *parallel*: both `GeneA` and `GeneB` influence `Phenotype`, but are on separate (parallel) paths;
2. *epistatic*: `GeneB` is epistatic to `GeneA` in a path for `Phenotype` with a given `Influence` (both genes are therefore on the same path and `GeneB` acts after `GeneA`);
3. *influences*: `GeneA` either excites or inhibits (`Influence`) the `Phenotype`;
4. *not influences*: `GeneA` does not influence the `Phenotype`.

As a final result of data analysis, GenePath derives genetic networks by satisfying abduced network constraints [8]. The network consists of nodes (genes and an outcome), and arcs that correspond to direct influences of genes on other genes and outcome. GenePath can derive models that include two types of influence: *excitation* (→) and *inhibition* (—|). As additional input, GenePath can also consider known parts of the network – a prior knowledge specified by the geneticist.

3 A Web-based Interface to GenePath

To make GenePath available to a wider audience, and in particular to experts and students in functional genomics, we have developed an interactive graphical user interface. The primary requirement was to design the interface that could be used on a wide variety of platforms. To avoid maintenance of different platform-specific versions, we have developed a web interface incorporating a server-based application that exports its user interface through a web browser. The resulting benefits are platform independence and low processing requirements on the client's side. No specific client-side installation procedure is required. A potential drawback is slower response time, especially at periods of intensive usage from several users, and at some points awkward user interface (compared to potential capabilities of stand-alone application), which is limited by HTML capabilities.

Our server-based application uses Active Server Pages (ASP) technology and Microsoft's IIS 5.0 web service. GenePath's abductive inference engine is implemented in SICStus Prolog (`http://www.sics.se`), and the communication between the engine and the interface is realized through SICStus's Visual Basic interface. Clickable images of genetic networks are generated using graph visualization software GraphViz (`http://www.graphviz.org`).

The user interface is quite intuitive and does not require special explanation. It follows a linear structure from the selection of a project, through definition of genetic data and background knowledge, all the way to the presentation of derived genetic networks and provision of explanation as to how the specific relations where found.

3.1 Data Entry

Data entry consists of specification of genes, outcomes, genetic experiments and background knowledge. The interface is straightforward, and supports savings and uploads to and from the client's system. Fig. 1 shows an entry screen for mutant data of *Dictyostelium*, where aggregation is observed as the outcome.

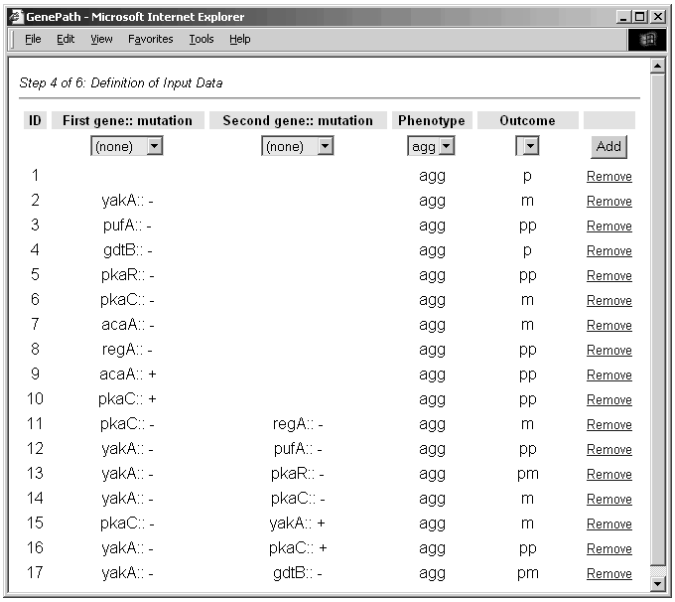

Fig. 1. A screen for definition and revision of genetic data

3.2 Presentation of Genetic Network and Constraints

For the data from Fig.1 GenePath finds a number of network constraints and three direct relations: yakA → pkaC, regA —| pkaC, and yakA —| pufA. Together with background knowledge (not shown here), the resulting genetic network is shown in Fig.2. The user can request the display of any type of constraints found by GenePath (*precedes* relations are displayed in the shown screenshot), or can click on any gene or edge of the network to display relevant experiments and constraints.

3.3 Explanation

The essential capability of GenePath's interface is the ability to provide explanation on abduced constraints. By clicking on the related evidence field of a specific constraint, the explanation is shown in a separate window. This includes the description of the pattern that was used to derive the constraint, and the experiments (or relations from background knowledge) that were involved.

For instance, Fig. 3 shows an explanation of the last constraint from Fig. 2: pufA is epistatic to yakA in a path for aggregation with a negative influence (inhibition). GenePath tells us that pufA was found to act after yakA in a path for aggregation because mutating these two genes separately leads to different outcomes, but mutating both genes at the same time leads to the same outcome as mutating only pufA. Therefore, it concludes that pufA blocks the influence of yakA; this is possible only if pufA is in the same path but after yakA.

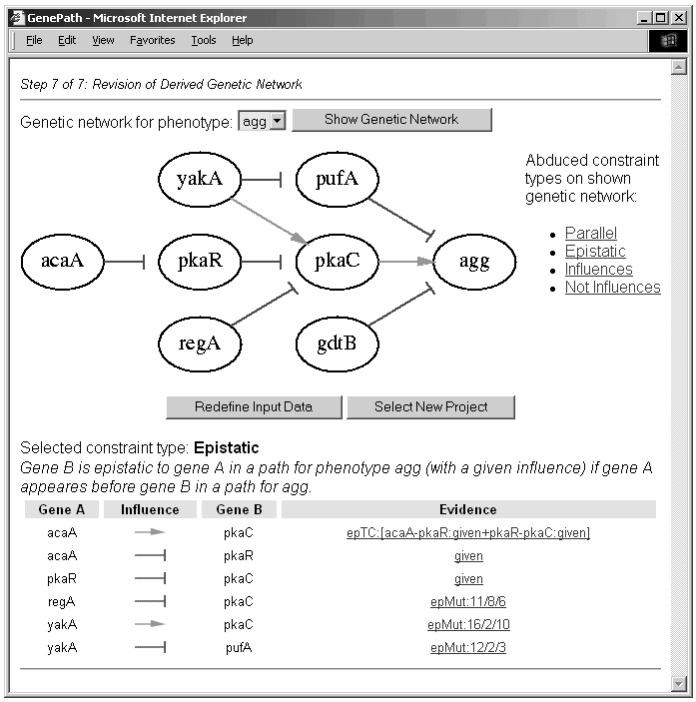

Fig. 2. Derived genetic network for cell aggregation and abduced constraints of type *precedes*

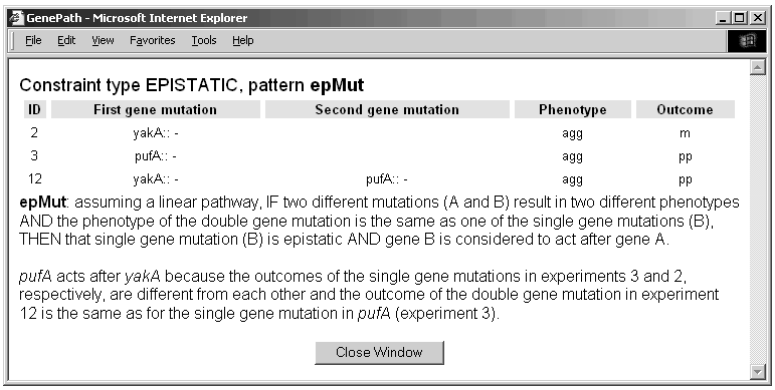

Fig. 3. Constraint explanation of the last constraint in Fig. 2: yakA —| pufA

Notice that GenePath's provision of explanations is possible because of its knowledge-based approach. The patterns that it uses for abduction of network constraints also define the language in which explanations are communicated. Since it was the domain experts that defined the patterns, GenePath's reasoning and explanation is bound to be transparent and comprehensible by biologists.

4 Discussion and Conclusion

GenePath derives genetic networks that are hypotheses over the specified set of experiments and prior knowledge. For instance, the network from Fig. 2 is slightly different than expected from the current state of domain knowledge, where genes yakA, pufA and pkaC are considered to be in linear relationship (yakA —| pufA —| pkaC). The reason that prevents GenePath to determine this relation is the absence of experimental data that would relate pufA and pkaC. These experimental data were obtained from biochemical experiments that were not included in table 1 [4, 5]. The advantage of GenePath is the ease and speed with which such observation is found. Furthermore, for the data shown in this paper, the expert biologists found that the constraints derived by GenePath were consistent with their knowledge, and that additional experiments can be further engineered based on what GenePath has found.

GenePath and its web-based interface are both evolving projects. We are currently extending their functionality by considering a combination of abductive inference and qualitative reasoning [8], allowing GenePath to propose and rank a set of potential networks rather than show a single network consistent with the constraints. Further extensions will also include experiment proposal, where GenePath will indicate for which genes the relations could not be established from the data, and will suggest the experiments that would resolve the resulting ambiguities in the genetic network.

The approach we are developing is intended for functional genomics community, particularly for students and researchers in the area, and will for that purpose be freely available on the Internet. Interested reader is welcome to check and use GenePath at `http://magix.fri.uni-lj.si/genepath`. The web site also provides for a number of test cases, including the one under the name “Project 3a: Dictyostelium Development (Aggregation Only)” described in this paper.

References

1. Bratko, I.: *Prolog Programming for Artificial Intelligence*. 3rd edition. Addison-Wesley (2001)
2. Flach, P.: *Simply Logical: Intelligent Reasoning by Example*. John Wiley & Sons (1994)
3. Hieter, P., Boguski, M.: Related Articles Functional genomics: it's all how you read it. *Science* (1997) 278(5338):601-2
4. Souza, G. M., da Silva, A. M., Kuspa, A.: Starvation promotes dictyostelium development by relieving PufA inhibition of PKA translation through the YakA kinase pathway. *Development* (1999) 126(14):3263-3274
5. Souza, G. M., Lu, S., Kuspa, A.: YakA, a protein kinase, required for the transition from growth to development in Dictyostelium. *Development* (1998) 125(12):2291-2302
6. Zhang, M. Q.: Large-scale gene expression data analysis: a new challenge to computational biologists. *Genome Res* (1999) 9:681-688
7. Zimmer, C.: The slime alternative. *Discover* (May 1998) 86-93
8. Zupan, B., Bratko, I., Demšar, J., Beck, J. R., Kuspa, A., Shaulsky, G.: Abductive Inference of Genetic Networks. *Proc. Artificial Intelligence in Medicine Europe*. Cascais, Portugal (2001)