GenePath: a System for Automated Construction of Genetic Networks from Mutant Data

*Blaz Zupan[1, 5], Janez Demsar[1], Ivan Bratko[1, 2], Peter Juvan[1], John A Halter[3], Adam Kuspa[4] and Gad Shaulsky[5, *]*

[1] University of Ljubljana, Faculty of Computer and Information Science and [2] Jozef Stefan Institute, Ljubljana, Slovenia, Departments of [3] PM&R and Division of Neuroscience, [4] Biochemistry and Molecular Biology and [5] Molecular and Human Genetics, Baylor College of Medicine, Houston, TX 77030, USA

ABSTRACT

Motivation: Genetic networks are often used in the analysis of biological phenomena. In classical genetics, they are constructed manually from experimental data on mutants. The field lacks formalism to guide such analysis, and accounting for all the data becomes complicated when large amounts of data are considered.

Results: We have developed GenePath, an intelligent assistant that automates the analysis of genetic data. GenePath employs expert-defined patterns to uncover gene relations from the data, and uses these relations as constraints in the search for a plausible genetic network. GenePath formalizes genetic data analysis, facilitates the consideration of all the available data in a consistent manner, and the examination of the large number of possible consequences of planned experiments. It also provides an explanation mechanism that traces every finding to the pertinent data.

Availability: GenePath can be accessed at http://genepath.org.

Contact: gadi@bcm.tmc.edu

Supplementary information: Supplementary material is available at http://genepath.org/bi-supp.

INTRODUCTION

Geneticists use mutations to investigate biological phenomena, because mutations alter the behavior (phenotype) of the system and reveal possible components of the biological process. Initially, mutations help define genes that participate in a biological process. Relationships between genes are then determined using combinations of mutations in two or more genes. Genetic networks that outline the details of a biological mechanism are constructed by integrating the relationships between pairs of genes. The effort required for ordering gene function is minimal compared to that required for obtaining the data, but the task becomes complicated when the data sets are large. We describe a software tool, GenePath, which automates the consideration of all the data in a consistent manner and allows geneticists to examine the possible consequences of planned mutations. GenePath processes experimental data and prior knowledge, constructs a genetic network and presents it as a graph. The output allows the user to examine the experimental evidence and the logic behind each relationship without becoming an expert in the specific problem.

GenePath infers genetic networks limited to non-cyclical graphs in which non-terminal nodes correspond to genes, terminal nodes correspond to biological processes, and arcs are labeled either “inhibits” or “excites”. The genetic logic used in GenePath is similar to that described for regulatory networks (Avery and Wasserman, 1992). In regulatory networks, signals are integrated through a cascade of gene products until they exert an effect on the biological process. Experiments consist of inactivation or excessive activation of genes such that the state of upstream genes becomes irrelevant to the phenotype. When mutations are made in two genes, the prevailing phenotype is defined by the mutation of the epistatic or downstream gene. This is different from the analysis of metabolic pathways or of developmental ‘dependent sequences’. In metabolic pathways, a mutation in an upstream node blocks the supply of metabolites to

downstream nodes, thus rendering downstream mutations irrelevant. In developmental pathways, a mutation in an upstream gene blocks the development of cells that would express downstream genes, thus making mutations in later genes ineffective.

In one application, used as an illustration in this paper, GenePath was applied to study a process that regulates the transition from growth to development in the social amoeba *Dictyostelium*. Upon starvation, the amoebae stop growing and develop into a multicellular fruiting body. Figure 1 describes a network that regulates that transition and we show that GenePath can reconstruct this and other networks from experimental data and from prior knowledge.

SYSTEM AND METHODS

Genetic Data

GenePath receives data in the form of phenotypes of single or double mutants. Table 1 lists genetic data for the aggregation of *Dictyostelium*, which will be used to illustrate the introduced concepts. The first experiment describes the wild-type phenotype and the other experiments describe mutations in one or two genes. In the mutants, genes are either inactivated (denoted by "-", *e.g.*, *regA*-, Table 1, experiment 7) or activated (denoted by "+", *e.g.*, *acaA*+, Table 1, experiment 8). The possible degrees of aggregation are: -, ±, +, ++. Degree "+" denotes wild-type aggregation, "++" excessive or rapid aggregation, "-" no aggregation and "±" reduced or delayed aggregation. GenePath typically considers qualitative phenotypes, but numerical values are also acceptable. The user must specify an ascending order of values, from weakest to strongest, from slowest to fastest, etc.

Prior Knowledge

Prior knowledge can also be included. In our example we included the following data:

1) *acaA* → *pkaC*(*acaA* excites *pkaC*; Pitt et al., 1992; Taylor et al., 1990)
2) *pkaR* –| *pkaC* (*pkaR* inhibits *pkaC*; Mutzel et al., 1987; Taylor et al., 1990)
3) *regA* → *pkaR* (Shaulsky et al., 1998)
4) *pufA* –| *pkaC* (Souza et al., 1999)

Relation "–|" denotes inhibition, *e.g. 2), pkaR* inhibits *pkaC*, and "→" denotes excitation, *e.g. 3), regA* excites *pkaR*.

Inference Patterns

The genetic logic in GenePath is defined through a set of inference patterns like "IF a certain combination of data is found , THEN a certain relationship between a gene and a biological process is hypothesized". The patterns belong to one of the following categories:

1. Influence: does a gene influence the biological process?
2. Parallelism: do two genes act in parallel paths of a genetic network?
3. Epistasis: does one gene act after another in the genetic network?

The patterns are described below with examples on the data set in Table 1.

Influence

These patterns relate genes to biological processes. They search for evidence that a gene influences a biological process and determine the influence type. GenePath includes two 'influence' patterns: **inf** and **infTC**.

inf: IF a mutation in a gene changes the phenotype relative to an otherwise identical strain, THEN the gene influences the biological process.

This pattern is straightforward and relates all the genes to the biological process in our example (Table 1). GenePath also determines the sign of the influence. If an activating mutation increases the phenotype, then the influence is positive and the gene "excites" the biological process (*e.g.*, Table 1, experiments 8, 9). The same applies if a gene inactivation decreases the phenotype (*e.g.*, Table 1, experiments 2, 5). The influence of a gene is negative if either the phenotype increases after gene inactivation (*e.g.*, Table 1, experiments 3, 4) or decreases after gene activation.

The second 'influence' pattern, **infTC**, defines relationships between genes and biological processes even in the absence of direct experiments. This pattern relies on the 'epistatic' relation (see below).

infTC: IF gene B is epistatic to gene A AND gene B influences the biological process, THEN gene A influences the biological process.

The pattern is applied iteratively: it finds genes that match the condition, asserts the relation into a data base, and repeats the process until no more new relations are found.

Parallelism

When genes act in parallel, their influence on a common downstream element is the integrated contribution of the influence of each gene alone. By finding parallel genes, GenePath determines that they cannot function in a single path of genetic network. The following pattern finds such genes:

parDiff: Two genes are in parallel genetic paths IF mutations in either gene have an effect on the biological process AND the phenotype of the double mutant is different from either mutation alone.

The genes *yakA* and *pkaR* match this pattern. They are considered to act in parallel because the phenotypes caused by the single gene mutations in (Table 1,

experiments 2, 4) are different from each other and from the phenotype of the double mutant (experiment 12).

Epistasis

The patterns for epistatic relations consider two genes and determine their order.

epMut: IF two different mutations (in genes A, B) result in two different phenotypes AND the phenotype of the double gene mutation is the same as the phenotype of the single gene B mutation, THEN that gene B is epistatic to gene A.

Notice that if gene B is epistatic to gene A then there is a directed path in the genetic network from A to B. The pattern **epMut** applies to several sets of experiments in Table 1. For instance, in experiments 5, 7, and 10, inactivation of *pkaC* reduces aggregation, inactivation of *regA* increases aggregation, and inactivation of both genes results in reduced aggregation, respectively. Consequently, GenePath concludes that *pkaC* is epistatic to *regA*. Similarly, GenePath finds that *pkaC* is epistatic to *yakA* (experiments 2, 9, 15) and *pufA* is epistatic to *yakA* (experiments 2, 3, 11).

GenePath also determines the sign of the influence (excitation or inhibition) between the two genes based on the sign derived from the "influences" relation: if the genes influence the phenotype in the same way, then the upstream gene excites the epistatic one, e.g., both *yakA* and *pkaC* excite aggregation, hence *yakA* → *pkaC*. Otherwise, the upstream gene inhibits the epistatic one, e.g., *regA* –| *pkaC* because *regA* inhibits aggregation.

The second epistasis pattern, **epTC**, defines relations based on other relations rather than on direct data. Like influence pattern **infTC**, **epTC** is applied iteratively.

epTC: IF gene B is epistatic to gene A AND gene C is epistatic to gene B, THEN gene C is epistatic to gene A.

Pattern **epTC** applies to three pairs of genes in our example (Table 1): 1) *acaA* → *pkaC* because *acaA* –| *pkaR* and *pkaR* –| *pkaC* (prior knowledge 1 and 2); 2) *regA* –| *pkaC* because *regA* → *pkaR* and *pkaR* –| *pkaC* (prior knowledge 2 and 3); 3) *yakA* → *pkaC* because *yakA* –| *pufA* (from the **epMut** pattern) and *pufA* –| *pkaC* (prior knowledge 4). The sign of an **epTC** relation is determined as in the relation **epMut**.

Construction of Genetic Networks

A genetic network hypothesized by GenePath consists of nodes and edges. The nodes represent genes or biological processes and the edges represent excitatory (→) or inhibitory (–|) relations. Notice that while consistent with a standard convention for drawing genetic networks, these symbols have different meaning than when used to denote epistasis and influence relations. GenePath constructs a genetic network by considering all the relations as constraints over the possible networks, and attempting to find a network that satisfies the constraints.

GenePath first checks for conflicts between the constraints. A typical conflict is a pair of genes that show both epistatic and parallel relations. Another conflict occurs if a gene influences the biological process and there is evidence for both negative and positive influences. Conflicts are reported to the user who may resolve them by assessing the reliability of the data or by performing additional experiments.

Next, GenePath considers the epistatic relations. It identifies pairs of genes with epistatic relations and examines their adjacency. Two genes are considered adjacent if GenePath cannot find other genes between them. For example, the following epistatic relations were found in the data: *acaA* → *pkaC*, *acaA* –| *pkaR*, *pkaR* –| *pkaC*. Is *acaA* adjacent to *pkaC*? Relation *acaA* → *pkaC* suggests that this may be possible, but *pkaR* is epistatic to *acaA* (*acaA* –| *pkaR*) and inhibits pkaC (*pkaR* –| *pkaC*), so *acaA* and *pkaC* are not adjacent. On the other hand, *acaA* and *pkaR* are adjacent because *pkaR* is epistatic to *acaA* and there is no evidence for intervening genes. Similarly, we find that *pkaR* and *pkaC* are adjacent, so we can infer a fragment of the network: *acaA* –| *pkaR* –| *pkaC*.

Finally, GenePath draws the hypothesized genetic network. It places genes and biological processes as nodes in a graph, drawing corresponding edges between adjacent nodes. Genes that are not followed by other gene are directly linked to the biological process with an edge that shows their influence. In our example, *pkaC* is the only terminal node. The network inferred by GenePath is as presented in Figure 2.

A BLIND TEST

GenePath was tested successfully on several data sets (see http://genepath.org/bi-supp), but in all cases we were aware of the desired genetic network. To test GenePath more stringently, we used a blind schema where one of the authors selected a published genetic problem, coded it and gave the data to the other authors who analyzed it using GenePath.

The data (Table 2) include 79 experiments with 16 genes. Each experiment consists of a mutation in zero, one, or two genes. The phenotype was either "+" or "-". Single mutant phenotypes are defined by the intersection of a specific row and the wild-type (WT) column, or vice versa. Double mutant phenotypes are defined by the intersection of the respective row and column.

Initially, GenePath revealed several epistatic relations that resulted in a cyclic path within a genetic network (*e.g.*, 1 → 3, 3 → 2, 2 → 17, 17 → 1). GenePath indicated that all the cycles involved gene number 2. No cycle-free networks were found. After removing the data for gene 2,

we obtained a conflict-free set of relations where all the genes influence the biological process through 54 epistatic relations. GenePath proposed a genetic network that is consistent with all the data (Figure 3).

The original publication from which Table 2 was extracted describes a network that regulates dauer larva formation in *C. elegans* (Riddle et al., 1981). The genetic network hypothesized by GenePath is identical to that proposed by the original authors, excluding gene number 2. That gene (*daf-2*) was also problematic in the original publication and was hypothesized to function in a parallel path that involved only some of the other genes (Riddle et al., 1981). Subsequent studies showed that *daf*-2 has additional functions that may have confounded the original analysis (Lin et al., 2001). Some of the original results are now considered inaccurate and a revised view of the process has been published (Thomas et al., 1993). In that regard, GenePath cannot detect erroneous data that are internally consistent.

The data of Riddle *et al*. represent a relatively large set of experiments. GenePath was able to handle the data and to propose a network within seconds. Overall, the results of the blind test demonstrate that GenePath performed its logical task correctly, and was capable of solving a complicated genetic problem and of calling the user's attention to special circumstances.

IMPLEMENTATION

GenePath's core is implemented in the Prolog programming language, and embedded in a server-based application in Visual Basic that provides for a web-based interface. Prolog (Programming in Logic), a declarative computer language that is often associated with development of Artificial Intelligence-based applications (Bratko, 2001), is effective in defining and reasoning with patterns (more in http://genepath.org/bi-supp). The web-based interface makes GenePath platform independent and easy to use. Data entry includes the definition of genes and biological processes, specification of prior knowledge and entry of experiments. The data can be saved to a local file for later use, revision, or dissemination.

A particular advantage of GenePath is explanation: clicking on any edge (arrows) reveals a list of experiments that provide evidence for the relation and text that explains the underlying logic. Clicking on any node reveals all the experiments that involve the selected gene and its relation to other genes and to the biological processes. Figure 4a gives an example of a GenePath results window (corresponding to Figure 2) and Figure 4b provides the explanation for the relation *yakA* –| *pufA*.

DISCUSSION

Utility

The most significant advantage of GenePath is its formalism: the program applies a fixed set of rules to all the data whereas manual use may lead to inconsistent application of the rules. For example, there are no formal rules that justified the decision to split the dauer larva regulatory genetic network in *C. elegans* as described in the blind test above, but the authors proposed an original solution that accounted for all the data (Riddle et al., 1981). Instead, GenePath called our attention to the problem with gene number 2 and presented a number of partial genetic networks that were consistent with all the data. Other such examples are given in the supplement (http://genepath.org/bi-supp).

GenePath analyzes the data and returns a genetic network in a fraction of the time required to perform that task manually. GenePath also alerts the user to conflicts that may otherwise be ignored and prompts the user to document the reasons for ignoring some of the data. The interface allows the user to explore the reason for each relation and facilitates the exploration of the network by non-experts. This feature may also be useful for teaching the principles of genetic analysis. GenePath can be used to test genetic models and to help design new experiments by entering new mutations along with possible phenotypes and finding which experiments would be the most informative. GenePath also allows researchers to document and communicate their data in a consistent manner.

GenePath handles classical genetic data, which consist of mutations in single or in multiple genes and the corresponding phenotypes. Normally these data sets contain a dozen or so genes, but GenePath was also developed in anticipation of the accumulation of vast amounts of genetic data. Work in *S. cerevisiae* has demonstrated the feasibility of generating hundreds of double mutants (Tong et al., 2001) and others demonstrated the feasibility of analyzing thousands of mutants in parallel (Ross-Macdonald et al., 1999; Winzeler et al., 1999). Such experiments are being performed in other organisms (Kuspa et al., 2001; Sucgang et al., 2000), so the need for automated methods for genetic network analysis is evident. For the data presented in this paper, GenePath constructed a network within 1 second of CPU time (Pentium IV, 900 MHz). We also tested GenePath on several large artificial data sets (see supplement) and found that GenePath effectively handles data on a hundred genes and several hundred experiments within 5 seconds, and data on 1000 genes and several thousand experiments within 40 minutes.

Limitations

GenePath proposes a single genetic network that accounts for the relations found in the data, but there may be many networks that are consistent with the data. Out of the plausible networks, GenePath proposes the one that orders the genes in a single path only if corresponding epistatic relations are found. Future versions of GenePath will propose a number of plausible networks and rank them according to expert-defined complexity measures.

It should be emphasized that GenePath is intended to construct a genetic network in much the same way as a geneticist would. Therefore GenePath mimics expert geneticist's reasoning about genetic data. For this purpose, GenePath uses rules (inf, infTC, parDiff, epMut, and epTC) whose formal definition closely follows the informal inference patterns actually used by the geneticists when manually constructing genetic networks. There are important mathematical questions regarding these inference rules. Are these set of rules logically sufficient and/or necessary? That is, do they suffice to derive from data *all* the gene-gene and gene-phenotype relations that are actually logically implied by the data? Are they a minimal (non-redundant) set of such rules? At the moment we do not have mathematical proofs to answer these questions. However, the geneticists who defined the patterns believe (based on their extensive use and experience) that these rules are sufficient in the abovementioned sense. We know that the set of rules is not minimal (rule parDiff is not necessary), but the geneticists find some redundancy useful as additional justification for conclusions inferred from data.

Currently, GenePath is only capable of pointing out conflicting constraints, the experiments that cause them and the conflict-free relationships. The user must evaluate the data and decide whether some experiments should be repeated or modified.

Many biological processes rely on feedback mechanisms to regulate the activity of their components. Feedback mechanisms appear in genetic networks as loops in which two or more genes regulate each other. The current version of GenePath does not address loops explicitly, but it enables the researcher to recognize potential loops if they occur in the data.

Genetic analysis is limited by the quality of the data it uses. GenePath was trained on a set of data that included mostly null alleles and a few selected constitutive alleles. Such mutations are usually the most simple to interpret, but they are not always available and not always the most informative. Mutants generated in genetic screens may involve a partial loss- or a partial gain-of-function, which may exhibit a variety of phenotypes. GenePath treats them as if they were null alleles or constitutive alleles and the user must address the partial effects of the mutations by different means.

The genetic method used in GenePath follows the logic of signaling pathways, whereas the logic of metabolic pathways and developmental pathways is usually reversed. It is easy to adapt the program to the solution of metabolic or developmental pathways by inverting the logical patterns, but the user must decide what type of genetic network is being analyzed. GenePath assists the user only by removing the need to analyze and document the data in a consistent manner.

Related work

The set of rules used in GenePath is widely used by geneticists but no other publications have stated these rules except for Avery and Wasserman (Avery and Wasserman, 1992). GenePath's novelty is in the formalization and automated application of these rules and in the public application of the program through the World Wide Web (http://genepath.org).

Computationally, GenePath borrows concepts like explicit encoding of knowledge, logic programming and utility of expert-based patterns in data analysis from Artificial Intelligence (AI) and performs abductive reasoning (see Kakas et al., 1998, and http://genepath.org/bi-supp) to find relations from the genetic data. While probably the best known AI system in genetics is an expert system for planning gene-cloning experiments in molecular genetics MOLGEN (Stefik, 1981), there are a number of contemporary systems that use some AI concepts and apply them in discovery of genetic networks. For instance, Friedman at al. (2000) use Bayesian networks to discover and Shrager et al. (2002) use heuristic search to revise genetic network, and Akutsu et al. (2000) infer genetic networks in the form of Boolean or qualitative networks. All mentioned systems derive genetic network from microarray data, and to the best of our knowledge GenePath is the only computer-based system to assist in classical genetic analysis. Like in GenePath, most contemporary systems infer networks which are directional and include both excitation and inhibition links. Compared to related work, GenePath is also unique in its explanation capabilities, where each finding can be traced back to experiments that support it.

ACKNOWLEDGEMENTS

We thank V. Lundblad, C. Shaw and K. Kibler for critical reading of the manuscript. This work was supported by a grant from the National Institute of Child Health and Human Development (P01 HD39691-01), by a grant from Slovene Ministry of Education, Science and Sport (J2-3387-1539) and by a travel grant from the National Academy of Sciences under the Collaboration in Basic Science and Engineering supported by Contract No. INT-0002341 from the National Science Foundation. G.S. is a

recipient of the Basil O'Connor research award from the March of Dimes Birth Defects Foundation (5-FY99-735).

REFERENCES

Akutsu, T., Miyano, S., and Kuhara, S. (200) Inferring qualitative relations in genetic networks and methabolic pathways. Bioinformatics 16, 727-34.

Avery, L., and Wasserman, S. (1992). Ordering gene function: the interpretation of epistasis in regulatory hierarchies. Trends Genet 8, 312-6.

Bratko, I. (2001). "Prolog Programming for Artificial Intelligence." Addison-Wesley.

Friedman, N., Linial., M., Nachman, I., and Pe'er, D. (2000). Using Bayesian networks to analyze expression data. J. Comp. Bio. 7, 601-620.

Kakas, A. C., Kowalski, R. A., and Toni, F. (1998) The role of Abduction in Logic Programming", in the Handbook in Artificial Intelligence and Logic Programming, Volume 5, (eds. D. Gabbay, C. Hogger and J. Robinson), pp. 235-324.

Kessin, R. H. (2001). "Dictyostelium - Evolution, cell biology, and the development of multicellularity." Cambridge Univ. Press, Cambridge.

Kuspa, A., Sucgang, R., and Shaulsky, G. (2001). The promise of a protist: the Dictyostelium genome project. Functional and Integrative Genomics 1, 279-293.

Lin, K., Hsin, H., Libina, N., and Kenyon, C. (2001). Regulation of the Caenorhabditis elegans longevity protein DAF-16 by insulin/IGF-1 and germline signaling. Nat Genet 28, 139-45.

Loomis, W., Dimond, R., Free, S., and White, S. (1977). Independent and dependent sequences in development of Dictyostelium. In "Eukaryotic microbes as model developmental systems." (D. H. O'Day and P. A. Horgen, Eds.), pp. 177-193. Dekker, New York.

Loomis, W. F. (1998). Role of PKA in the timing of developmental events in Dictyostelium cells. Microbiol. Mol. Biol. Rev. 62, 684.

Mutzel, R., Lacombe, M. L., Simon, M. N., de Gunzburg, J., and Veron, M. (1987). Cloning and cDNA sequence of the regulatory subunit of cAMP-dependent protein kinase from Dictyostelium discoideum. Proc. Natl. Acad. Sci. USA 84, 6-10.

Pitt, G. S., Milona, N., Borleis, J., Lin, K. C., Reed, R. R., and Devreotes, P. N. (1992). Structurally distinct and stage-specific adenylyl cyclase genes play different roles in Dictyostelium development. Cell 69, 305-315.

Riddle, D. L., Swanson, M. M., and Albert, P. S. (1981). Interacting genes in nematode dauer larva formation. Nature 290, 668-71.

Ross-Macdonald, P., Coelho, P. S., Roemer, T., Agarwal, S., Kumar, A., Jansen, R., Cheung, K. H., Sheehan, A., Symoniatis, D., Umansky, L., Heidtman, M., Nelson, F. K., Iwasaki, H., Hager, K., Gerstein, M., Miller, P., Roeder, G. S., and Snyder, M. (1999). Large-scale analysis of the yeast genome by transposon tagging and gene disruption. Nature 402, 413-8.

Shaulsky, G., Fuller, D., and Loomis, W. F. (1998). A cAMP-phosphodiesterase controls PKA-dependent differentiation. Development 125, 691-699.

Shrager, J., Langley, P., and Pohorille, A. (2002) Guiding revision of regulatory models with expression data. Proc. Pacific Symposium on Biocomputing 7:486-497.

Souza, G. M., da Silva, A. M., and Kuspa, A. (1999). Starvation promotes Dictyostelium development by relieving PufA inhibition of PKA translation through the YakA kinase pathway. Development 126, 3263-3274.

Souza, G. M., Lu, S. J., and Kuspa, A. (1998). YakA, a protein kinase required for the transition from growth to development in Dictyostelium. Development 125, 2291-2302.

Stefik, M. (1981). Planning with constraints (MOLGEN: Part1). Artificial Intelligence 16, 111-140.

Sucgang, R., Shaulsky, G., and Kuspa, A. (2000). Toward the functional analysis of the Dictyostelium discoideum genome. J Eukaryot Microbiol 47, 334-9.

Taylor, S. S., Buechler, J. A., and Yonemoto, W. (1990). cAMP-dependent protein kinase: framework for a diverse family of regulatory enzymes. Annu. Rev. Biochem. 9, 971-1005.

Thomas, J. H., Birnby, D. A., and Vowels, J. J. (1993). Evidence for parallel processing of sensory information controlling dauer formation in Caenorhabditis elegans. Genetics 134, 1105-17.

Tong, A. H., Evangelista, M., Parsons, A. B., Xu, H., Bader, G. D., Page, N., Robinson, M., Raghibizadeh, S., Hogue, C. W., Bussey, H., Andrews, B., Tyers, M., and Boone, C. (2001). Systematic genetic analysis with ordered arrays of yeast deletion mutants. Science 294, 2364-8.

Ullrich, A., Shine, J., Chirgwin, J., Pictet, R., Tischer, E., Rutter, W. J., and Goodman, H. M. (1977). Rat insulin genes: construction of plasmids containing the coding sequences. Science 196, 1313-9.

Winzeler, E. A., Shoemaker, D. D., Astromoff, A., Liang, H., Anderson, K., Andre, B., Bangham, R., Benito, R., Boeke, J. D., Bussey, H., Chu, A. M., Connelly, C., Davis, K., Dietrich, F., Dow, S. W., El Bakkoury, M., Foury, F., Friend, S. H., Gentalen, E., Giaever, G., Hegemann, J. H., Jones, T., Laub, M., Liao, H., Davis, R. W., and et al. (1999). Functional characterization of the S. cerevisiae genome by gene deletion and parallel analysis. Science 285, 901-6.

Table 1. Experimental data on *Dictyostelium* aggregation

Exp #	Genotype	Aggregation {-, ±, +, ++}
1	wild-type	+
2	yakA-	-
3	*pufA-*	++
4	*pkaR-*	++
5	*pkaC-*	-
6	*acaA-*	-
7	*regA-*	++
8	*acaA+*	++
9	*pkaC+*	++
10	*pkaC-*, *regA-*	-
11	*yakA-*, *pufA-*	++
12	*yakA-*, *pkaR-*	+
13	*yakA-*, *pkaC-*	-
14	*pkaC-*, *yakA+*	-
15	*yakA-*, *pkaC+*	++

Table 2. Genetic data for the double-blind test

	WT	1	2	4	7	8	11	14
WT	-	+	+	+	+	+	+	+
3	-	-	+	-	-	-	-	-
5	-	-	+	-	-	-	-	-
6	-	+	+	+	-	-	-	-
10	-	+	+	n/a	+	+	+	+
12	-	-	+	-	-	-	-	-
16	-	-	-	+	-	-	-	-
17	-	+	-	+	-	-	-	+
18	-	+	+	+	+	-	-	+
20	-	-	-	+	-	-	-	-

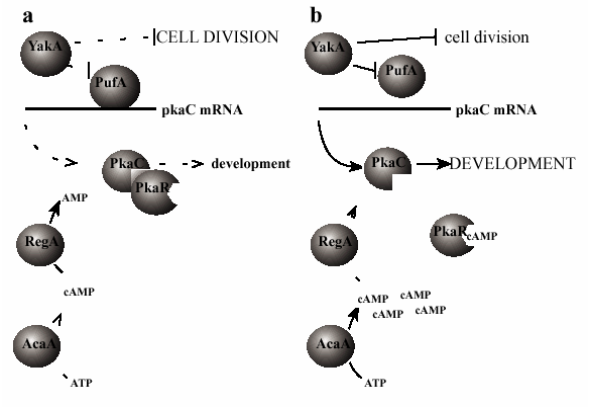


Figure 1. The transition from growth to development in *Dictyostelium*

a. During growth, YakA is inactive and cell division is not inhibited (Souza et al., 1998). The PufA protein binds the pkaC mRNA and inhibits the translation of PkaC, the catalytic subunit of PKA (Souza et al., 1999). Production of cAMP by the adenylyl cyclase AcaA is also low due to low levels of *acaA* gene expression (not shown). The phosphodiesterase RegA degrades cAMP to 5'AMP and as a consequence the regulatory subunit of PKA (PkaR) can associate with the PkaC protein and inhibit its protein kinase activity (Mutzel et al., 1987; Shaulsky et al., 1998; Taylor et al., 1990). As a result, the activity of PkaC is low in growing cells and the entry into development is inhibited.

b. Upon starvation, activation of YakA leads to inhibition of cell division and to inhibition of PufA activity (Souza et al., 1999). Consequently, pkaC mRNA is free to be translated and high levels of PkaC protein are produced. Production of cAMP by AcaA is also increased, mainly due to induction of the *acaA* gene expression (Pitt et al., 1992). Upon binding to cAMP, the PKA regulatory subunit PkaR loses its ability to bind and inhibit the catalytic subunit PkaC (Mutzel et al., 1987). PkaC is activated and development begins.

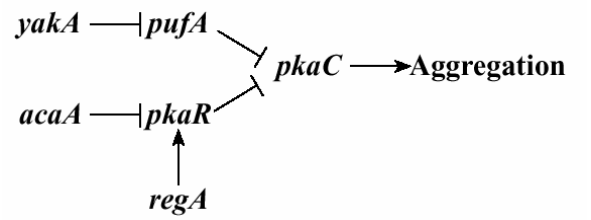


Figure 2. A regulatory network for *Dictyostelium* aggregation. The network was derived by GenePath from the data shown in Table 1. See text for detail.

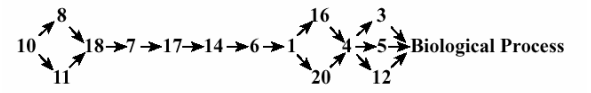
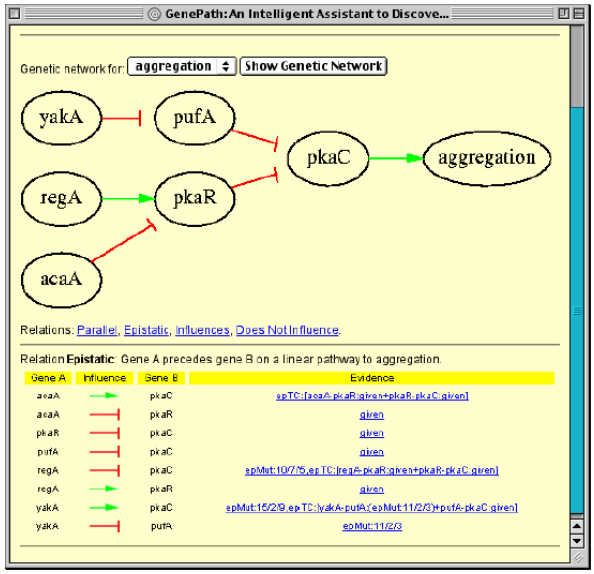


Figure 3. A blind test of GenePath. The network was derived by GenePath from the data shown in Table 2. See text for detail.



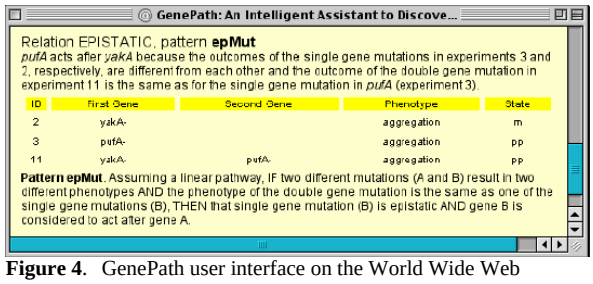


Figure 4. GenePath user interface on the World Wide Web
a. A GenePath results window with a list of epistatic relations for the *Dictyostelium* aggregation data set.
b. A GenePath window with an explanation for the relation *yakA* –| *pufA*.