SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

SCHOOL OF BIO AND CHEMICAL ENGINEERING

DEPARTMENT OF BIOINFORMATICS

UNIT – I - SBIA5301 – SYSTEMS BIOLOGY

MULTISCALE COMPUTATIONAL MODELING

In the postgenomic era, researchers seek to focus their attention to studying and analyzing biological networks and pathways by the use of multiscale computational modeling techniques. A model can be viewed as a representation of a biological system, where the representation can comprise a set of differential equations [8], a set of first-order logic clauses [9], and so on. Biological models that incorporate multiple scales such as time and space or multiple timescales may be viewed as multiscale models [10]. Chapter 2 gives an in-depth account of mathematical and computational models in systems biology.

Development of efficient and effective computational methodologies to perform modeling, simulation, and analysis of complex biological processes is a challenging task. Traditionally, mathematical and computational models have been developed by considering a single scale. However, it is now feasible to incorporate multiple scales in the process of model building due to recent advances in computational power and technology. Generally, multiscale models are constructed by using sophisticated techniques including numerical methods and integration approaches. Multiscale model of the heart [11, 12] is a well-known example of an application of these modeling techniques.

Multiscale computational modeling and simulation methods are showing promising results in the field of oncology. The development of three-dimensional multiscale brain tumor model by Zhang et al. [13] is an attempt in this direction. The dynamics of tumor growth were simulated by using an agent-based multiscale model where microscopic scale, macroscopic scale, and molecular scale were incorporated in the *in silico* model. In micro-macroscopic environment, a virtual brain tissue block was represented by points in three-dimensional lattice. The lattice was divided into four cubes that illustrated the behavior of chemotactically acting tumor cells. The chemotaxis distribution of transforming growth factor alpha (TGFα), glucose, and oxygen tension were illustrated in a set of mathematical equations. It was observed that the amount of TGFα and glucose was chemoattractant, and diffusion of glucose occurred at a constant rate. In order to incorporate molecular scale, epidermal growth factor receptor (EGFR) gene–protein interaction network model [14] was used in conjunction with cell cycle module. The authors used a simplified EGFR network that comprised of EGFR and TGFα genes. The mathematical model of EGFR gene–protein network was represented as a set of differential equations.

The authors utilized the cell cycle model presented in Tyson and Novak [15] and Alacron et al. [16]. The implementation of the software systems was carried out by combining in-house code with an agent-based software tool, namely, MASON (`http://cs.gmu.edu/ eclab/projects/mason/`). In order to study and analyze tumor growth and spread, 10 simulations were performed. The results demonstrated an increase in tumor volume with respect to time, where the relationship between tumor volume and time was not linear. There was a sharp increase in volume growth at later time intervals. The study found that migrating and proliferating cells exhibited a dynamic behavior with respect to time. Furthermore, the cells caused spatiotemporal tumor growth. The results showed that the number of migrating cells was greater than the number of proliferating cells over time, where the high concentration of phospholipase C gamma (PLCγ) might be the key factor behind the phenomenon. In summary, the study demonstrated a successful construction of multiscale computational model of the complex multifaceted biological process. However, the approach is not free from shortcomings as described below:

- A simple EGFR network was used.
- Clonal heterogeneity within tumor was not examined.

It has been found that the distribution of tumor cells is not homogeneous, and the cells exhibit heterogeneous patterns. Techniques that account for clonal heterogeneity of tumor cell populations can be vital to analyze and study the development of cancerous diseases. Furthermore, clonal heterogeneity can strongly impact the design of effective therapeutic strategies. Therefore, many studies examined heterogeneity in tumors [17, 18]. Zhang et al. [19] extended their multiscale computational modeling technique [13] to investigate the clonal heterogeneity by incorporating genetic instability. The extended model included doubling time of cell and cell cycle. Other parameters such as cell–cell adhesion were also considered so that the strength of the chemoattractants' (TGFα, oxygen tension, and glucose) impact on cancer cells adhesion and rate of cell migration could be investigated. The authors used Shannon's entropy for the quantification of tumor heterogeneity. Shannon entropy in this context can be calculated as follows: Let c_i denote the occurrence of clone i in the tumor, the entropy is given by $\sum_i c_i \ln(c_i)$, where the higher values of Shannon's entropy represent more clonal heterogeneity.

The results of the study showed an increase in tumor total volume over time, where the tumor was categorized into three regions on the basis of the distance between it

and the nutrient source. It was observed that there was a general increase in the values of Shannon's entropy for all the three regions. However, there was highest clonal heterogeneity in the region closest to the nutrient source at early time stages where the region exhibited a homogeneous pattern at later stages. The study inferred that cancer could spread faster due to clonal heterogeneity as compared to homogeneous cell populations in tumor.

The complexity of the mechanisms of development and morphogenesis establishes a need to design effective and efficient computational techniques to investigate and analyze the biological process. In a recent study, Robertson et al. [20] presented a multiscale computational framework to investigate morphogenesis mechanisms in *Xenopus laevis*. Mammalian cells share similarities with *X. laevis* in terms of signaling network and cell behavior. A multiscale model was constructed by integrating an intercellular signaling pathway model with the multicellular model of mesendoderm migration. The authors implemented Wnt/β-catenin signaling pathway model that was presented by Lee et al. [21], whereas an agent-based approach was applied

and the nutrient source. It was observed that there was a general increase in the values of Shannon's entropy for all the three regions. However, there was highest clonal heterogeneity in the region closest to the nutrient source at early time stages where the region exhibited a homogeneous pattern at later stages. The study inferred that cancer could spread faster due to clonal heterogeneity as compared to homogeneous cell populations in tumor.

The complexity of the mechanisms of development and morphogenesis establishes a need to design effective and efficient computational techniques to investigate and analyze the biological process. In a recent study, Robertson et al. [20] presented a multiscale computational framework to investigate morphogenesis mechanisms in *Xenopus laevis*. Mammalian cells share similarities with *X. laevis* in terms of signaling network and cell behavior. A multiscale model was constructed by integrating an intercellular signaling pathway model with the multicellular model of mesendoderm migration. The authors implemented Wnt/β-catenin signaling pathway model that was presented by Lee et al. [21], whereas an agent-based approach was applied
to build mesendoderm migration model. In order to simulate mesendoderm cells' migration, it was viewed that each cell comprised of nine sections, where each section was modeled as an agent. Mesendoderm migration was facilitated by the use of fibronectin extracellular matrix substrate. The study found that fibronectin gradient was a key factor behind the cellular movement. It was also observed that polarity signals [22] might be important for mesendoderm migration and morphogenesis. The simulations also demonstrated the importance to keep the cadherin binding strength in balance with the integrin binding strength. Although the study establishes the efficacy of multiscale computational methodologies to studying morphogenesis, the proposed approach may not be computationally attractive for large-scale simulations.

Physiome project [12] is well known for the development of multiscale modeling infrastructures. Given that standard modeling languages are useful for sharing biological data and models, three markup languages, namely, CellML (`http://www.cellml.org/`), FieldML, and ModelML, have been developed in the project. CellML [23] is characterized by its ability to capture three-dimensional information regarding cellular structures. It can also incorporate mathematical knowledge and metadata. FieldML, a related language, is known for its incorporation of spatial information. The third systems biology modeling language, namely, ModelML, is characterized by its ability to encode physical equations that illustrate complex biological processes. The efficacy of the languages was established by building multiscale heart models [12].

It has been found that same input, to constituent parts of a system, can produce different outputs. Such variations may be produced by factors including alterations in the concentration of system's components. It is desirable to design techniques and methods that can provide robustness to variations. Shinar et al. [24] presented a robust method by exploiting molecular details. The authors coined the term "input–output relation" for the association between input signal strength and output. The study investigated the input–output relation in bacterial signaling systems.

PROTEOMICS

Proteomics, the study of proteins, is viewed crucial to analyze and understand biological systems, as protein is the building block of life. Mass spectrometry (for details see Chapter 17) is a well-known proteomics technology that is showing a huge impact on the development of the field of computational systems biology. Several recent studies have identified the significant role of proteomics techniques in solving complex biological problems [25–27].

Proteomics methods and data can be useful for the reconstruction of biological networks. Recently, Rho et al. [28] presented a computational framework to reconstruct biological networks. The framework is based on the use of proteomics data and technologies to build and analyze computational models of biological networks. It is termed as integrative proteomic data analysis pipeline (IPDAP). IPDAP incorporates a number of network modeling and analysis tools. The component tools of IPDAP can be applied to reconstruct biological networks by fusing different types of proteomics data. The successful application of IPDAP to different cellular and tissue systems demonstrated the efficacy and functionality of the framework.

In another study, Zhao et al. [29] investigated signal transduction by applying techniques from optimization theory and exploiting proteomics and genomics data. They formulated the network identification problem as an integer linear programming problem. The proteomics (protein–protein interaction) data were represented as weighted undirected graph, where the nodes and the edges represented proteins and interaction between pair of proteins, respectively. The results of the study confirmed the efficacy of the approach in searching optimal signal transduction networks from the data.

Cell cycle comprises a series of ordered events by which cell replication and division take place. Studying cell cycle regulation provides useful insights in cancer growth and spread. The relationship between cell cycle and cancer has been a focus of many studies [30, 31]. In Sigal et al. [32], a proteomics approach was applied to investigate cell cycle mechanisms. The approach is based on the use of time-lapse microscopy to study protein dynamics. The study identified cell cycle-dependent changes in protein localization, where 40 percent of the investigated nuclear proteins demonstrated cell cycle dependence. Another challenging problem is to find patterns of polarized growth in cells where such growth is viewed as an important process in organisms. In order to investigate the biological problem, Narayanaswamy et al. [33] conducted a study by using budding yeast as the model system. The proposed computational method is based on the use of microarray image analysis and a machine learning technique, namely, naive Bayes algorithm. The study found 74 localized proteins including previously uncharacterized proteins and observed novel patterns of cell polarization in budding yeast.

In a recent study [34], a computational technique is presented for predicting peptide retention times. The method is at the intersection of two machine learning approaches, namely, neural networks and genetic algorithms. In order to predict the retention times, an artificial neural network is trained and the predicted values are further optimized by using a genetic algorithm. The method was successfully applied to *Arabidopsis* proteomics data.

COMPUTATIONAL SYSTEMS BIOLOGY AND AGING

Aging is a complex phenomenon that has not been well understood. In aging, we witness gradual diminishing/decreasing functions at different levels, including organs and tissues. Cell division has been viewed as a key process in aging since long [35, 36]. Recently, de Magalhaes and Faragher [37] have elucidated that aging might be affected by variations in cell division. Hazard rates and nutrition may be the key factors that influence the longevity of cellular organisms [38]. There are a number of theories that describe how aging occurs. Kirkwood [38] listed five different theories that are as follows:

- Somatic mutation theory
- Telomere loss theory
- Mitochondrial theory
- Altered proteins and waste accumulation theory
- Network theory

Aging has been extensively studied in *Caenorhabditis elegans* (nematode), mice, humans, and fruit flies. A number of genes that extend organisms' life span have been discovered. Several studies on aging found that genetic mutations could increase longevity [39–41]. Furthermore, aging genes with their associated pathways may influence the variations in aging between different species but may not have any affect on the differences in aging within a particular specie [42]. Gene expression and pathway analysis can provide useful means to identify aging-related similarities and differences between various species [43], where the efficacy of DNA microarray technology, in studying aging, is significant [44]. In a recent study on aging, DNA microarray experiments were utilized to show that aging in *C. elegans* is influenced by GATA transcriptional circuit [45].

Advances in computational systems biology have led to the development of tools and methods for solving highly complex problem of aging. For example, Xue et al. [46] addressed the key issue regarding aging by applying an analytic method to human/fruit fly protein–protein interaction network, namely, NP analysis [47]. The method is based on the identification of active modules in network, where the chosen module comprised of protein–protein interaction subnetwork between genes that show (positive or negative) correlation during aging. The application of the method to human brain aging identified four modules. Among these modules, the two showed transcriptionally anticorrelation with each other. The other two modules comprised of immunity genes and translational genes, respectively. In order to study correlation between genes in other species during aging, the method was applied to fruit fly interactome. The results of the study showed that in addition to two transcriptionally anticorrelated genes modules, there were two other modules that demonstrated such anticorrelation. On the basis of these findings, the authors suggest that only a few modules are associated with aging. The other key result of the study is the identification of the influence of module connecting genes on aging.

In another study, Garan et al. [48] presented a computational systems biology framework for studying neuroendocrine aging. The framework allows fusion of heterogeneous data from different disciplines such as endocrinology, cell biology, genetics, and so on. The method can be effective in identifying underlying relationship between the components that define aging.

Machine learning provides useful approaches and techniques to conduct studies on aging. In Swindell et al. [49], a number of machine learning methods were used to predict mouse life span. Twenty-two learning algorithms were applied to the problem, where the results demonstrated usefulness of support vector machines (SVMs), stabilized linear discriminant analysis, and nearest shrunken centroid in solving the problem, hence establishing the efficacy of machine learning technique for aging research. Agent-based modeling techniques have also been used to understand the biological processes of aging. The study published by Krivenko and Burtsev [50] is indicative of the success of such approaches for aging related studies. The authors applied their technique to simulate evolution and studied important factors including kin recognition and aggression.

Analysis of pathways for aging can also facilitate the understanding of complex diseases such as cancer. The probability of the occurrence of a cancer can be substantially lowered by downregulating the aging pathways [39]. Recently, Bergman et al. [51] investigated longevity genes. They conducted an extensive study by using more than 1200 subjects. On the basis of system-based analysis, the authors recommend that the investigation of genetic pathways can lead to the development of strategies that may regulate age-related diseases and disorders.

COMPUTATIONAL SYSTEMS BIOLOGY IN DRUG DESIGN

Millions of people are suffering from fatal diseases such as cancer, AIDS, and many other bacterial and viral illnesses. Computational systems biology approaches can provide a solution to the key issue that is how to design lifesaving and cost-effective drugs so that the diseases can be cured and prevented. Pharmaceutical companies view that systems-based computational techniques will be highly useful in designing effective therapeutic drugs [52–54]. Furthermore, advanced and sophisticated methods will accelerate drug discovery and development. In 2007, FDA approved only 17 new drugs [55] and approximately 50 drugs in 2008 (`http://www.fda.gov/`).

It is believed that the association between systems-based biological methods and drug design is age-old. Herbal drugs were developed by observing the diseases; hence, today's drug design has been (directly/indirectly) influenced by such early attempts [56]. Computational systems biology approaches may revolutionize therapeutic intervention in clinical medicine [2]. Effective systems-based drug design techniques can be developed by exploiting the knowledge of the robustness of biological systems [57].

An overview of a number of computational methods' (Petri nets, cellular automata techniques, hybrid methods, pi calculus, agent systems, and differential equations-based methods) application to the task of drug design can be found in Materi and Wishart [52].

Identification of novel drug targets in diseases is a key problem. In order to solve such problems, Chu and Chen [58] recently presented a systems-based approach for the identification of apoptosis drug targets. The selection of the drug targets by utilizing the approach can be viewed as a multistage discovery process. In the first stage, a protein–protein interaction network is constructed by a number of datasets and online interactome databases. In the second stage, a stochastic model of protein–protein interactions is constructed. In order to refine the model, false protein interactions are removed by utilizing an information theoretic measure, namely, Akaike's information criterion to microarray data. Finally, drug targets are identified by conducting a network-level comparison between normal and cancer cells.

Transcription factors-based methods can play an important role in devising an effective therapeutic and preventive interventions strategy for diseases. In Rosenberger et al. [59], the role of activating transcription factor 3 (ATF3) was investigated for murine cytomegalovirus (MCMV) infection. Mouse was used as the model system. The study demonstrated negative regulation of interferon-gamma (IFN-γ) expression caused by ATF3 in natural killer cells. The mice that had zero ATF3 exhibited high resistance to MCMV infection.

In another study, Nelander et al. [60] introduced a computational systems biology methodology for the prediction of pathway responses to combinatorial drug perturbations or drug combinations. The method is based on the use of multiple input–output model. Given that the linear models are not able to capture crucial information required for the task at hand, the authors presented nonlinear multiple input–output model. The approach was applied to analyze perturbations in MCF7 human breast carcinoma cells, where a number of compounds including rottlerin, rapamycin, and and so on were selected as perturbants. The leave-one-out cross-validation results showed the efficacy of the method.

Genetic causes of diseases can provide information that is crucial to design effective therapeutic approaches. A network that illustrates the association between diseases and their related genes can be highly informative. The human disease network presented in Goh et al. [61] is an attempt in this direction. The graph theoretic framework is based on the construction of a network to analyze and investigate the association between phenotypes and disease genes. In the constructed bipartite graph, one set of nodes represents genetic disorders and the second set denotes known disease genes in human genome. The edge between the disease and a gene represents the mutation in gene caused by the disease. The network provides a means to study novel patterns of gene disease associations.

Screening toxic compounds is a key issue in drug design and development. In Amini et al. [62], a novel computational methodology was introduced as an accurate means of predicting toxicity of compounds. The technique integrates two machine learning approaches, namely, SVMs [63] and inductive logic programming (ILP), and is termed support vector inductive logic programming (SVILP). The method works by obtaining a set of rules from an ILP system, hence mapping the compounds into relational ILP space. The induced rules are then applied to compute the similarity between two compounds by the use of a novel kernel function. The function, given by an inner product in relational ILP space, is a weighted sum over all the common hypothesized rules. The ILP kernel is used in conjunction with SVMs to compute toxicity. The authors applied their method to a diverse and broad ranging toxicity dataset, namely, DSSTox [64]. The effectiveness of the method was established by using a cross-validation experimental methodology to predict the toxicity of the compounds. The results of the study confirmed the efficacy of the method for drug design and development. In Lodhi et al. [65], the method is extended to classify mutagens and recognize protein folds. The extended method learns a multiclass classifier by using a divide-and-conquer reduction strategy that divides multiclasses into binary groups and solves each individual problem by inducing an SVILP. The extended multiclass SVILP was successfully applied to classify compounds.

The database storing detailed kinetic knowledge can be a useful resource as it can provide information that is required to build models of biological processes. In order to provide such a knowledge base, a database of kinetic data, namely, KDBI, has been developed [66]. The database contains various types of data, including protein–protein interactions and protein–small molecule interactions. It includes 19,263 records, where 2635 entries belong to protein–protein interactions and 11,873 records contain information regarding protein–small molecule interactions. The database also comprises ordinary differential equations-based pathways models.

SOFTWARE TOOLS FOR SYSTEMS BIOLOGY

In this section, we will very briefly describe software tools that are designed for modeling, simulating, and analyzing complex biological processes. Bioconductor is a project that provided a number of useful tools for conducting systems biology-based studies. The design of effective infrastructure is crucial for the development of efficient and user-friendly tools. Software infrastructures may be developed by using only a basic computer language and generator (a software tool) [67]. Chapter 15 provides an in-depth description of a text mining tool for systems biology. Table 1.1 summarizes a number of software packages for studying and investigating biological systems.

SQUAD [68] is an example of modeling tools for systems biology. It constructs dynamic models of signaling networks, where the unavailability of kinetic data do not hinder its performance. The underlying methodology of the systems is based on the integration of Boolean and continuous modeling techniques. The implementation is written in Java, whereas C++ has been used to code algorithms for the computation of steady states. SQUAD supports a number of input formats, including NET (text file), MML (xml file), and SMBL (systems biology markup language). The system performs simulations as follows: It takes as input a directed graph representing the structure of the network. The steady states of the graph are identified by

Table 1.1 Software tools for systems biology

Tools	Biological systems	Input format	Platform
	Modeling		
SQUAD	Signaling and regulatory networks	XML, MML, and NET	Windows and Linux
CellNetAnalyzer	Metabolic, signaling, and regulatory networks	Network Composer and ASCII	All platforms (approximately)
BioTapestry	Signaling and regulatory networks	CSV and tabular	Linux, Mac, and Windows
	Sensitivity Analysis		
SBML-SAT	Signaling, regulatory and metabolic network	SBML	Linux, Mac, and Windows
	Visualization		
Cytoscape	Molecular interaction networks	MS Excel, SIF, and so on	All platforms (approximately)
CellProfiler	Cell images	DIB	Linux, Mac, and Windows

using a Boolean algorithm. Then, a dynamic model is constructed. Finally, a user can perform simulations. SQUAD has a user-friendly graphical interface and can be downloaded from http://www.enfin.org/dokuwiki/doku.php?id=squad:start.

CellNetAnalyzer [69] is a related software tool for modeling and analyzing biological process. It can be applied to analyze signaling, regulatory, and metabolic networks. The software tool is implemented in MATLAB, and C has been used to code some underlying techniques. The input data can be provided to CellNetAnalyzer by using Network Composer or ASCII file. It is available at http://www.mpi-magdeburg.mpg.de/projects/cna/cna.html.

BioTapestry [70] is another biological modeling tool. It can perform analysis and modeling of large biological networks. Linux, Windows, and Mac are supported platforms. BioTapestry is available at http://www.biotapestry.org/.

Sensitivity analysis is an important aspect of computational modeling for systems biology. SBML-SAT [71] performs sensitivity analysis of biological systems, and the systems are represented in the form of ordinary differential equations. It incorporates and implements a number of well-known sensitivity analysis techniques. Windows, Mac, and Linux are supported platforms. SBML-SAT is implemented in MATLAB, where the input data need to be coded in SBML format. It is available at http://sysbio.molgen.mpg.de/SBML-SAT/.

We now briefly describe Cytoscape [72] that facilitates the visualization and analysis of biological networks. It also allows data integration. The supported input formats are delimited text files, MS Excel, SIF (simple interaction format), SMBL, GO (gene association), and so on. It enables the identification of active modules in biological networks. Cytoscape also allows export of network structures as images in different formats. Cytoscape is available at http://www.cytoscape.org/.

The development of CellProfiler [73, 74] is an attempt to study complex biological processes by using image analysis software packages. The tool comprises two components, namely, CellProfiler and CellProfiler Analyst. The images are processed by using CellProfiler. CellProfiler Analyst is applied to analyze the processed data produced by CellProfiler. The tool can analyze hundreds and thousands of images. It is characterized by its capability of recognizing nonmammalian cells and quantification of phenotypes. It supports processing and analysis of multidimensional images and can perform illumination correction and cell identification by using standard and advanced methods. The tool is implemented in MATLAB and is available for Windows, Unix, and Mac platforms. The software tool is available at http://www.cellprofiler.org/.

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

SCHOOL OF BIO AND CHEMICAL ENGINEERING

DEPARTMENT OF BIOINFORMATICS

UNIT – II- SBIA5301 – SYSTEMS BIOLOGY

UNIT II MODELS IN SYSTEMS BIOLOGY The Parameter Problem and the Meanings of Robustness - Models as Dynamical Systems - Continuous Models - Discrete Models - The Parameter Problem – Parameter phobia - Measuring and Calculating - Counter Fitting - Beyond Fitting - The Landscapes of Dynamics - Qualitative Dynamics - Steady State Attractors of ODE Models - The Meanings of Robustness Parameter Biology - Robustness to Initial Conditions - Robustness in Reality - Structural Stability - Classifying Robustness Rule-Based Modelling and Model Refinement -A Simple Cascade - A (Natural) Computing Perspective on Cellular Processes - Cell Cycle and Breast Tumor Growth Control

MODELS AS DYNAMICAL SYSTEMS

Two broad directions have emerged in systems biology. The first, "omics," initiated by new technologies such as the microarray [17], relies on inferring causality from correlation in large datasets (see, for instance, Sieberts and Schadt [18]). To the extent that models are used, they are statistical in character. The second direction, which might be called "mechanistic" systems biology, has been less visible but has deeper historical roots [7–11]. The resulting models specify molecules, cells, and tissues and their interactions based on what is known or believed to be true. It is with the latter type of model that we will be concerned here. The subtleties of causal analysis are well discussed elsewhere [19].

Most mechanistic models in systems biology can be regarded as some form of *dynamical system*. A dynamical system describes the *states* of a biological system and how these states change in time. It can be abstractly visualized as in Figure 2.1 as a *state space*, upon which is imposed a temporal dynamics: Given a particular state as an *initial condition*, the dynamics define the *trajectory* taken over time from that starting point. Not all models take this form. For instance, constraint-based models represent systems at steady state and have no explicit representation of time [20]. We focus here on models that do.

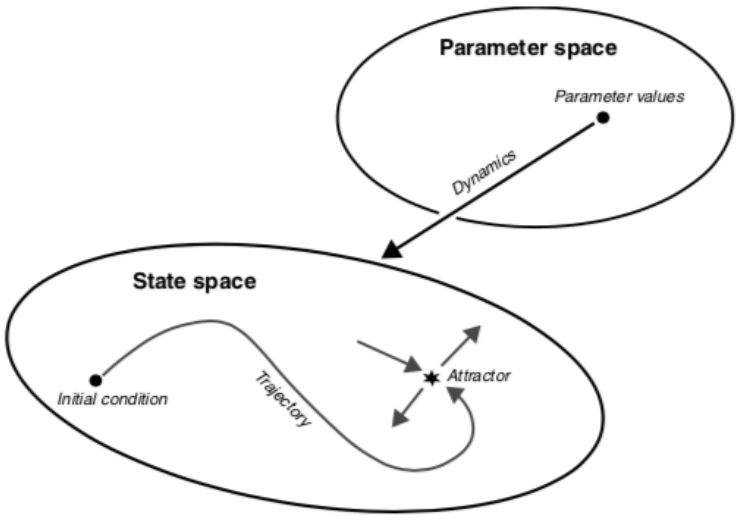

Figure 2.1 Dynamical system. A point in parameter space, given by a set of parameter values, defines the dynamics on the state space. If the system is prepared in an initial condition, then the dynamics typically lead to an attractor, pictured here as a star. Common attractors are steady states or periodic orbits but they can be much more complex [46]. Note that some trajectories leave the attractor, indicating that it is unstable, as discussed in Section 2.4.1. The parameter and state spaces are pictured as abstract sets. For ODE models, they usually correspond to Euclidean spaces, $\mathbb{R}^k$, of some dimension k but for other kinds of models the state space can be infinite dimensional (PDEs or stochastic models) or not have any linear structure (discrete models).

THE PARAMETER PROBLEM

Biological systems have many "moving parts," whose collective interactions produce the physiology or phenotype of interest. Two general strategies have emerged to model this complexity. One seeks to bring the model's assumptions close to reality by embracing the details of components and interactions. The resulting models are *thick*, with many states and more parameters. The other strategy moves in the opposite direction and seeks to abstract the essentials from the details, giving rise to *thin* models with fewer parameters. Despite parochial assertions to the contrary, both strategies have provided biological insight; their pros and cons are discussed in the companion paper to this [2]. In both cases, but most especially with thicker models, the problem arises of determining parameter values in a way that maintains credibility in a model's conclusions. The importance of this problem has tended to be obscured in the literature for several reasons. On the one hand, it is easier to assert (particularly to an experimental audience) "This model accounts for the data" than "This model, with these parameter values, accounts for the data." The latter formulation invites awkward questions as to why those parameter values were chosen and not others. (One might have included "initial conditions" along with parameter values but since the initial conditions are values of state variables, they share the same level of measurability and are, therefore, usually easier to determine than parameter values.) Even if editors and reviewers are aware of the problem—and it seems they are mostly not—they are generally disinclined to ferret about in the Supplementary Information, to which graveyard such technical details are usually consigned. Finally, such a variety of approaches have something to say about the problem that it is hardly surprising to find confusion as to best practice. Here, we emphasize the significance and centrality of the parameter problem by contrasting different disciplinary perspectives of it.

Parameterphobia

Parameters are anathema to physicists, who take the view expressed in the quotation from von Neumann that, with enough parameters, any behavior can be modeled. Of course, von Neumann was joking: a weighted sum of increasing functions with positive weights (parameters) can never fit a decreasing function, no matter how many parameters are used. (See Section 2.4.2.1 for a more relevant example.) However, the truth behind the joke distills a long tradition of modeling the inanimate world on the basis of the fundamental laws of physics. Biology, while founded entirely upon these laws, is not modeled in terms of them. Molecular or cellular behavior is not deduced from Schrödinger's equation. At best, a model may be based on chemical principles such as the law of mass action. At worst, it may rely on some ad hoc guess that is only tenuously related to specific biological knowledge, let alone an underlying molecular mechanism. We have, in such cases, no systematic methodology for avoiding parameters.

While physicists are familiar with parameters and keep them firmly in their place, computer scientists (at least those of a theoretical disposition) are less acquainted with them. The discrete models used in theoretical computer science, like finite automata or Turing machines, have no parameters [34]. (They may have labels but these are passive adornments that do not effect the rate of state transitions.) When discrete models are parameterized, they transmogrify into Markov chains, whose properties are more commonly studied elsewhere than in computer science. In consequence, computer science has had little to say about the parameter problem.

Measuring and Calculating

Ideally, parameter values should be independently measured. In practice, our limited ability to make quantitative measurements of molecular states makes this difficult if not impossible for many parameters. Even when parameters have been measured, the conditions may have been sufficiently different as to raise doubts as to the relevance of the measurements. *In vitro* values, for instance, may differ substantially from those *in vivo*, while *in vivo* measurements themselves may require very careful interpretation [41]. Nevertheless, such measurements as do exist are often useful for initial analysis. Molecular dynamics (MD) calculations—arising from atomic-scale models—can now provide illuminating explanations of intramolecular behavior [42]. Certain kinds of parameters, such as binding constants, might be calculated from such MD models. Since these calculations are limited largely by computational power, it would be unwise to bet against them in the long run, but it seems unlikely that they will yield a systematic approach anytime soon. They will, in any case, be limited to only certain kinds of parameters and to molecules whose atomic structures are well understood.

Counter Fitting

Engineers are accustomed to building thick models with many parameters—of chemical reactors or combustion chambers, for instance—and determining parameter values by fitting to quantitative data [16]. This is the strategy most widely adopted in systems biology when sufficient data of the right kind are available. The development of nonlinear optimization algorithms has made parameter fitting easy to undertake but has also concealed its dangers. These take several forms. The structure of a model may render it nonidentifiable *a priori*: It may not be possible, even in principle prior to any data fitting, to determine certain parameter values. Even if a model is identifiable, the fitting process itself may need to be carefully examined. The reported optimum may be only local. Even if a global optimum is found, there may be several parameter sets that yield roughly similar optimal values. In other words, the energy landscape underlying the optimization may be undulating with many optimal valleys rather than a broad funnel leading to a single optimum. A classic example is that of fitting a sum of two exponentials; see, for instance, Figure 4.6 of Lakowicz [43].

The second and more serious danger in model fitting brings us back to the broader significance of von Neumann's quip. How is a model to be rejected? The answer "when there are no parameter values that fit the data" would not have satisfied von Neumann because, in his view, a model that is complex enough may fit all manner of data. In other words, the rejection criterion is inadequate. As we will see in Section 2.4.2.1, the behavior of biochemical models is more subtle than this: models with arbitrary many parameters may sometimes have the simple qualitative behavior shown by Eq. (2.2). The core issue may be restated in terms of explanatory power. A model does not explain the data to which it is fitted; the process of fitting already incorporates the data into the model.

Of course, parameter fitting is widely used in other areas of science. An X-ray crystal structure, for instance, is obtained by fitting an atomic model to diffraction data, with many free parameters (bond angles, bond lengths, etc.). In such cases, independent cross-validation is used [44]. The data are partitioned into two sets: "test" data and "working" data. Parameters are determined by fitting on the working data. Having been fitted, they are used to account for the test data. If they do, the model is accepted; if not, it is rejected. Hodgkin and Huxley used a similar strategy for their famous model of the action potential in the squid giant axon [8]. The parameters were fitted in independent experiments on each of the three ion channels. Once fitted, the model, with those parameter values, was shown to numerically reproduce the time course of the action potential. Another strategy is to use wild-type data as working

data and mutant data to test it by computationally mimicking the effect of the mutation [45]. As these examples make clear, a model's explanatory power comes from being able to account for data to which it has not been fitted.

Merely showing that quantitative data can be accounted for with some choice of parameter values can be such an effort, particularly with thick models, that it is often regarded as sufficient in itself. While this is easy to get away with, at least at present, it is not a good foundation for a new discipline.

Beyond Fitting

Determining a specific set of parameter values and accounting for novel data is only part of the parameter problem. We have a general suspicion of models that are fine-tuned, for which some parameters require precise values. They are not "robust." (Much the same argument is made about unstable steady states; see Section 2.4.1.) Robustness is a good feature, so the argument goes, because there are always errors, often substantial errors, in measuring and fitting data. Related systems might also be expected to show qualitatively similar behavior but not have quite the same parameter values. If a model can be shown to be robust to changes in parameter values, then one can be more confident in drawing conclusions from it despite such uncertainties. There may also be properties of a model that are robust to variation in certain parameter values, like temperature compensation in circadian oscillators. Identifying such properties may yield biological insight; see Section 2.5.3. Aside from such robustness, which we will discuss further in Section 2.5, there may not always be sufficient quantitative data, or data of the right type, to fit all parameter values. The available data may, for instance, not be numerical but qualitative, as in developmental patterns. Finally, models can also be used in an exploratory way to understand how to think about a system in the first place, prior to any determination of parameter values. In all these cases, it becomes important to know how the model's behavior varies as a function of parameter values. This is the broader aspect of the parameter problem. To address it, a more qualitative view of dynamical systems becomes necessary.

THE LANDSCAPES OF DYNAMICS

Qualitative Dynamics

Although the general ideas outlined in this section apply to most forms of dynamical system, they are best understood for ODE models [23, 46]. Figure 2.2 illustrates, in a simple case, the kind of behavior to be expected of a model similar to example (2.1), in which

$$\frac{\mathrm{d}x}{\mathrm{d}t} = f(x;a), \tag{2.3}$$

where $x \in \mathbb{R}^n$ is a vector of state variables, $a \in \mathbb{R}^m$ is a vector of parameters, and $f : \mathbb{R}^n \to \mathbb{R}^n$ is the vector rate function expressing the balance between production

and consumption of each x_i. Biological state variables are frequently non-negative (concentrations, for instance) and the state space may then be taken to be the non-negative orthant of $\mathbb{R}^n$. For any given set of parameter values, the trajectory starting from a given initial condition will typically converge upon an *attractor*: a limited region of the state space within which trajectories become confined. For instance,

the trajectory may reach a steady state, as in example (2.1), or a periodic orbit, as in models of the cell cycle [23], circadian rhythms [47], or developmental clocks [48]. Chemical systems can also have more complex attractors and exhibit behaviors like bursting and chaos [49], which may have some biological role in the excitable tissues found in cardiac and neural systems [50]. A dynamical system may have several different attractors for a given set of parameter values. A familiar instance in systems biology is bistability [23, 51, 52], in which a dynamical system has three attractors, consisting of two stable steady states and one unstable steady state (Figure 2.2(c)). In this case, different initial conditions may reach different attractors and each attractor will have its own *basin of attraction* consisting of those initial conditions that lead to it. The state space breaks up into multiple disjoint basins of attraction, each leading to a unique attractor.

The geometry of a basin of attraction reveals something of the dynamics leading to the corresponding attractor. For instance, a steady state is stable if its basin of attraction has the same dimension as that of the ambient state space (dimension 2 for the two stable states in Figure 2.2(c)). If its dimension is lower, then moving away from the attractor along one of the missing dimensions leads outside the basin of attraction and toward some other attractor. This is the case for the saddle point in Figure 2.2(c) for which the basin of attraction has dimension 1. The argument is made that an unstable steady state is never found experimentally because random perturbations ("noise") would destabilize it. Stable states are "robust" to such perturbation. Consequently, a steady state of a model that is claimed to represent some observed behavior should always be checked to be stable. However, if only a few dimensions among hundreds are missing from a basin of attraction, then it may be possible for the system to linger in the corresponding steady state for an appreciable time, relative to the noise timescales in the system, before becoming destabilized. Our experience of high-dimensional systems is still too limited to know how significant this might be.

Steady State Attractors of ODE Models

Chemical Reaction Network Theory

2.4.2.1 Chemical Reaction Network Theory

Example (2.1) has only a single parameter region and only a single attractor—a stable steady state—for all parameter values in that region. Remarkably, more complex models may still exhibit similar behavior. This emerges from Feinberg's chemical reaction network theory (CRNT) [61]; see Gunawardena [62] for an overview and other references. CRNT applies to the ODE model coming from a network of chemical reactions by applying the principle of mass action. It associates with such a network a nonnegative integer called the "deficiency", which does not depend on the values of the parameters but only on the underlying network of reactions. The deficiency is the dimension of a certain linear subspace, reflecting one of the key insights of CRNT: Behind the nonlinearity of mass-action kinetics, there exists a remarkable degree of hidden linearity [62]. Under reasonable conditions, deficiency zero networks behave like example (2.1): Provided constraints are respected (see Section 2.5.2 for an explanation of constraints), there is a single parameter region and only a single stable steady state for all parameter values in that region [61, 62]. This theorem is important because it shows that thick models, with many parameters, may nevertheless have simple qualitative dynamics. One cannot always fit an elephant! Having said that, the "deficiency zero theorem" is too restricted to be widely used in systems biology, where parameter values have typically been found to influence the qualitative dynamics. Recent developments in CRNT may be more relevant [63] and the full implications of CRNT for systems biology remain to be worked out.

THE MEANINGS OF ROBUSTNESS

Robustness is one of the themes to have emerged in systems biology [72–75] and it is particularly relevant to the parameter problem. Unfortunately, it is also one of those concepts whose wide usage has not been matched by precise definition. Robustness means, broadly, that some property of the system remains the same under perturbation. To make this precise, it is necessary to say what the property is, in what sense it remains the same, and what kinds of perturbations are being considered. The property might be the overall qualitative dynamics of a system, in which case "remaining the same" could mean that the number and type of attractors and the connectivity and shape of the trajectories remain the same under perturbation. Alternatively, the property could be a quantitative function evaluated on an attractor, like the period of a periodic orbit. In this case, "remaining the same" could mean that the property remains quantitatively unchanged under perturbation ("exact robustness") or that it only changes by a limited amount ("approximate robustness"). As for perturbations, at least three different kinds can be distinguished: changes to parameter values, changes to initial conditions, and changes to the functional form that describes the dynamics (i.e., the f in Eq. (2.3) for an ODE model). These perturbations have distinct mathematical and biological implications. We will discuss the first two as preparation for reviewing some influential studies of robustness and then return to the third.

Parameter Biology

Consider an ODE model derived by the principle of mass action from a network of biochemical reactions. In this case, the parameters are rate constants of various kinds: association rates, disassociation rates, catalytic rates, and so on. Such rates are, hopefully (see the next paragraph), intrinsic features of the corresponding proteins and would not be expected to change except through alterations to their amino acid sequences. This could happen on an evolutionary timescale, so that different species may have different parameter values, but this would not be expected to happen in different cells of the same organism or tissue or clonal population of cells in cell culture. The situation could be different in a polyclonal population, such as a tumor or a natural population of outbred organisms, in which there could be substantial genetic polymorphism. Depending on which loci exhibit polymorphism and how it affects protein function, this genetic variation could give rise to rate constant variation between different cells or different organisms.

(A caveat is essential here. Rate constants are not solely determined by intrinsic features of a protein. They also depend on the ambient conditions in the cell—temperature, pH, and other ionic strengths—as well as, potentially, posttranslational modifications such as disulfide bridges or glycosylations, or the presence of accessory molecules such as chaperones or scaffolds, none of which might have been included in a model. The reductionist approach commonly used in systems biology, in which the properties of a system are deduced from its components, is always at risk of the system biting back: The properties of the components may depend on that of the system [2]. To put it another way, the boundary of a system has to be drawn somewhere, with the implicit assumption that what is outside the boundary is irrelevant to the behavior inside. Such assumptions tend to be taken for granted until they fail.)

Robustness to Initial Conditions

If the property thought to be robust is associated with an attractor, such as a steady state, then its robustness to initial conditions would seem to follow from the stability of the attractor, in the sense discussed in Section 2.4.1. However, it is often the case that the dynamics satisfy additional constraints. For instance, an enzyme suffers no net change in concentration in any reaction that it catalyzes. If it is not being otherwise synthesized or degraded, then its total concentration remains constant at all times. Similarly, if a substrate exists in many states of modification—multisite phosphorylation, for instance—and is also not synthesized or degraded, then its total concentration remains constant. (Note that these constraints are linear in the state variables; nonlinear constraints may also be possible.) If there are k independent constraints, they confine the dynamics to lie within a subspace of dimension $d = n - k$, where n is the dimension of the ambient space. The state space thereby becomes divided into "slices" of dimension d, each corresponding to a set of constraint values (Figure 2.4). Within each slice, the dynamics behave as they did in Figure 2.1, with attractors, basins of attraction and stability, as appropriate to an ambient space of dimension d (not n). However, its qualitative character can change with the constraint values. Hence, the constraint space also becomes divided into regions, within each of which the dynamics in the corresponding slices remain qualitatively similar (Figure 2.4).

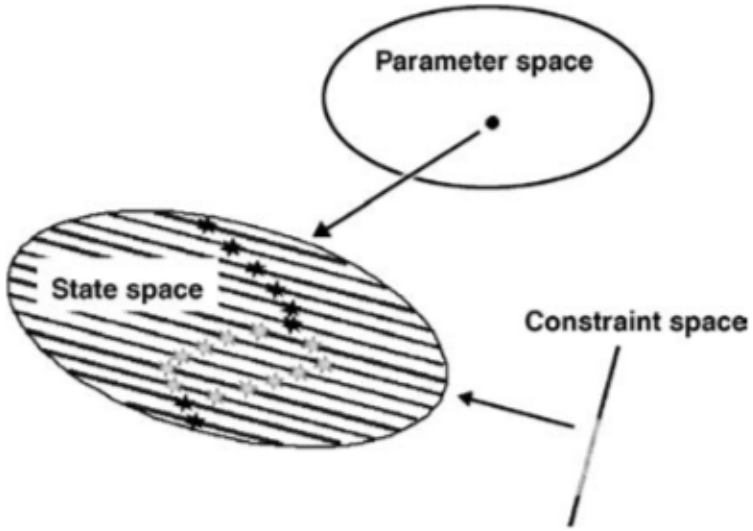

Figure 2.4 Dynamical system with constraints. The state space becomes divided into "slices," represented by the straight lines, each slice corresponding to a set of constraint values, represented by a point in the space of constraints. Note that if the invariants are nonlinear, then the slices may be curved spaces. The dynamics are confined within the slices. If an initial condition is chosen within a slice, then the trajectory remains within that slice for all time; trajectories never cross between slices. The dynamics within a slice can have attractors, represented by stars, and other features as described in Figure 2.1 but their qualitative character can change as the constraints vary, as illustrated by the appearance and disappearance of attractors.

Robustness in Reality

With this background, let us review some particularly interesting and influential demonstrations of robustness in different biological systems.

Signaling in bacteria is typically implemented by two-component systems consisting of a sensor kinase coupled to a response regulator protein [79]. The sensor autophosphorylates in response to a signal, using ATP as the phosphate donor. It then transfers the phosphate to the response regulator, which initiates the signaling response, by, for instance, stimulating gene transcription. In some two-component systems involved in homeostasis—such as the EnvZ/OmpR system that regulates osmolarity in *Escherichia coli*—the sensor also catalyzes the dephosphorylation of the response regulator. This unusual bifunctional mechanism has been studied in several models [80–82], whose general conclusion is that the mechanism enables the amount of phosphorylated response regulator at steady state to be constraint robust with respect to changes in the total amounts of sensor and response regulator. The initial analysis by Russo and Silhavy using Michaelis–Menten kinetics [81], which provided the first indication of this robustness, was subsequently refined using mass-action kinetics by Batchelor and Goulian [80]. Their analysis showed approximate constraint robustness when the amount of sensor kinase is much less than the amount of response regulator, which, indeed, corresponds well to *E. coli's* normal operating regime. In their accompanying experimental analysis, they varied the total amounts of EnvZ and OmpR and found good agreement with their model. Shinar et al. incorporated a further element into the mechanism by noting that in certain bifunctional two-component systems [82, Table 1], including the EnvZ/OmpR system in *E. coli*, ATP acts as a cofactor in the dephosphorylation of the response regulator. Their model for this shows exact constraint robustness of the amount of phosphorylated response regulator, with respect to changes to the total amounts of sensor, response regulator, and ATP, provided the amount of response regulator remains above a threshold. These predictions were also borne out by experiment.

E. coli has also been a model bacterium for the study of chemotaxis. It moves by rotating its multiple flagella. Rotation in one direction brings the flagella into alignment, allowing the bacterium to "run" in a straight line. Rotation in the other direction drives the flagella apart, causing the bacterium to "tumble" and randomly reorient its direction. By regulating its tumbling frequency, the bacterium can efficiently seek out nutrients and escape poisons (chemotaxis) in environments that lie outside its control. Because *E. coli* is so small, it has to sense changes in ligand concentration over time, not space. It has been found to adapt its sensitivity to such changes across a remarkably broad range of background concentrations. Unraveling the mechanism behind this has been a triumph of systems biology [83].

Structural Stability

Robustness with respect to functional variation—perturbing the f in Eq. (2.3)—has not been as widely utilized as the kinds of robustness described above. However, it was the basis for a remarkable historical episode that still has resonance for us today. Waddington's distillation of biological dynamics inspired the distinguished French pure mathematician René Thom to develop a mathematical framework for describing it [95]. Thom made two general assumptions. First, that the dynamics arose from descending down a gradient, so that $f(x;a) = -\nabla g(x;a)$, where $\nabla = \sum_{i=1}^{n} \partial/\partial x_i$ is the gradient operator. Waddington's epigenetic landscape has just such a gradient dynamics but for Thom the assumption arose from technical necessity rather than analogy and, in his case, the parameters play a key role. In gradient dynamics, steady states correspond to minima of the gradient function, g, which provides a crucial simplification. Second, Thom assumed that, in the absence of detailed knowledge about the underlying molecular mechanisms that gives rise to g, it was reasonable to focus on *structurally stable* behaviors; that is, those behaviors that remained qualitatively the same if the function g was perturbed, $g \to g + h$, where h is "small." Under these assumptions, Thom proved that, for small numbers of parameters ($m \leq 5$), there were only finitely many—in fact, just 11—different types of structurally stable bifurcations [96, Chapter 7]. Note that the state space can be of any dimension. Furthermore, most bifurcations that have been studied tend to depend on only a few parameters, with the others playing only a background role. Hence, in practice, the restriction to $m \leq 5$ is not limiting.

Classifying Robustness

One reason why robustness has attracted such attention is that it may be a biological design principle [74]. This is an appealing idea, but to make sense of it, robustness needs to be precisely defined and grounded in the kind of careful experiments discussed in Section 2.5.3. As we have shown, there are different types of robustness, which may be classified according to which aspect of the dynamical system is changed.

- *Type I: Dynamical Stability*. Robustness to change of initial conditions within a fixed set of constraint values.
- *Type II: Constraint Robustness*. Robustness to change of constraint values.
- *Type III: Parametric Robustness*. Robustness to change of parameter values.
- *Type IV: Structural Stability*. Robustness to change of the dynamical function.

No doubt there are others. As noted in Section 2.5.1, the interpretation of these mathematical properties depends crucially on the biological context that is being modeled. Robustness could be quantified if we could estimate the size and shape of various regions in high-dimensional spaces: basins of attraction, constraint regions, and parameter regions. Many studies can be seen as attempts to do this by random sampling [84, 99]. Lack of space precludes a discussion of robustness trade-offs [74, 104] and new methods of global sensitivity analysis [54, 55]. Kitano has remarked on the need for a theory of biological robustness [105]. The dynamical systems framework outlined here may provide a basis for this.

RULE BASED MODELING AND MODEL REFINEMENT

Rule-based modeling is an effective way of handling the explosive combinatorics of biological networks. The use of partial objects in describing molecular interactions means that only the necessary conditions for a rule are specified and not the complete chemical entities taking part in a reaction. This leads to descriptions that are easier to set up and more compact. Networks of substantial scale can be described without having to reduce the combinatorics of the system—as other approaches must.

An important aspect of the rule-based approach is its agility, as one can easily modify rules to incorporate new knowledge or test different assumptions. A special and rather frequent case is when one wishes to replace a rule with ones imposing stronger conditions. This process is called *refinement*, and we approach it in this study both from the practical and the theoretical point of view.

There are various reasons why one would like to use refinement:

- One wants to understand how the activity of a rule varies with its application contexts
- One realizes that more conditions are necessary than previously thought
- One more subtly wishes to evolve the behavior of the current system

The notion of behavior-preserving, or neutral, refinement commands an analysis of the possible symmetries of partial complexes. Here, we need a rigourous algebraic theory to see through the intricacies caused by symmetries. Incidentally, the problem of neutral refinement is one of a family of problems that is well-studied in the theory of concurrent systems, usually under the catch phrase of "behavioral equivalence." The form of equivalence we are looking for here is especially strong, since it should hold irrespective of the other rules defining the dynamics of the model.

The material is organized as follows. We begin with a brief introduction to the Kappa language (Section 4.1). Next, we present several examples (Section 4.2) of refinements. We have, in particular, a somewhat lengthy example that shows how refinements can be used to evolve complex behavior from simple systems. By introducing mutant variants of agents that alter the behavior of a single rule, it is possible to change dramatically and in unexpected ways the outcome of a pathway (Section 4.2.2).

Once we are reassured that the notion of refinement is actually useful, we turn to the second part, namely, the mathematical development of rule refinement. An algebraic version of (a mild simplification of) Kappa is introduced (Section 4.3). This is framed in basic category theory, which allows us to make use of existing mathematical techniques. Previous work in this area developed a framework for homogeneous rule refinement, where agents of the same type had the same sets of sites [1]. The framework developed here is much more general and introduces the notion of addresses to access specific agents in partial complexes (Section 4.4). This enables us to model a much larger class of rule refinements, and an example is given of a model that could not have been dealt with previously. We end by deriving a general formula for neutral refinement and show that the stochastic transition system underlying the rule set is unchanged.

The following is self-contained. Nevertheless, readers might want to consult earlier Kappa references on a concrete example of the agility of rule-based modeling [2], the use of debugging methods based on abstract interpretation [3], the development of techniques for large-scale stochastic simulation [4], or the study of statistical asymptotic properties of simple Kappa networks [5].

A Simple Cascade

In order to introduce our notation for rules and agents and demonstrate the notion of refinement in a first simple case, we start with an elementary cascade. This type of biological circuitry occurs frequently in actual pathways (e.g., see [8]).

In our example, we have one kinase S, covalently modifying another kinase X, which, in turn, modifies some third agent Y. Each agent type is supposed to have a single site, and the sites of X and Y hold an internal state of either u (unphosphorylated) or p (phosphorylated); one says, X and Y are active when they are phosphorylated. To keep things simple, the model does not include any mechanism to deactivate X or Y.

4.2.1.1 The Rules The interactions between S and X are defined by the following rules:

$$S(i),\, X(s_u) \rightarrow S(i^1),\, X(s_u^1),$$
$$S(i^1),\, X(s^1) \rightarrow S(i),\, X(s),$$
$$S(i^1),\, X(s_u^1) \rightarrow S(i^1),\, X(s_p^1).$$

In this rule set, a binding is represented by a shared exponent, for example, $S(i^1), X(s^1)$ represents a binding between the S and the X agents via their respective i and s sites. The first rule in the triplet specifies the conditions for such a binding to take place: one needs the sites i and s to be free and one also needs the site s to have a specific internal state u, indicated as a subscript s_u. One might say that S is 'smart' in so far as it does not bind a target that is already modified, that is, of the form $X(s_p)$. The second rule represents the unbinding of the two molecules. Contrary to the first one, this rule does not depend on the s site of X being in a particular internal state. The ability to not have to specify the entirety of the context in which an event can be triggered—which we alluded to earlier, and which is sometimes called the "don't care, don't write" convention—already shows here in a very simple form. The third rule represents the activation of X, that is, the change of X's internal state from u to p.

A second and similar rule triplet defines the interactions of X and Y:

$$X(s_p), Y(s_u) \to X(s_p^1), Y(s_u^1),$$
$$r := X(s^1), Y(s^1) \to X(s), Y(s),$$
$$X(s^1), Y(s_u^1) \to X(s^1), Y(s_p^1).$$

This rule set differs from the previous one only in that the X agent is required to have a phosphorylated s site in order to bind a Y agent, as stipulated in the first rule. This ensures that the first half of the cascade happens before the second and, in particular, that Y cannot be activated if there is no S signal.

CELL CYCLE AND BREAST TUMOR GROWTH CONTROL

In this section, we show how the computational paradigm introduced in Section 5.5 can be adapted in order to model important cellular processes. In particular, we show how it is possible to model the processes concerning cell cycle and breast tumor growth.

It is well-known that the life of human beings is marked by the cycling life of its constitutive cells. It goes through four repetitive phases: `Gap 1` (`G1`), `S`, `Gap 2` (`G2`), and `M`. `G1` is in between mitosis and DNA replication and is responsible for cell growth. The transition occurring at the restriction point (called `R`) during the G1 phase commits a cell to the proliferative cycle. If the conditions that enforce this transition are not present, the cell exits the cell cycle and enters a nonproliferative phase (called `G0`) during which cell growth, segregation, and apoptosis occur. Replication of DNA takes place during the synthesis phase (called `S`). It is followed by a second gap phase responsible for cell growth and preparation for division. Mitosis and production of two daughter cells occur in the `M` phase. Switches from one phase to the next one are critical checkpoints of the basic cyclic mechanism, and they are under constant investigation [20, 21].

Passage through these four phases is regulated by a family of cyclins[1] that act as regulatory subunits for the cyclin-dependent kinases (Cdks). Cyclins' complex activates Cdks, with the aim to promote the next phase transition. Such activation is due to sequential phosphorylations and dephosphorylations[2] of the key residues mostly located on each Cdk complex subunit. Therefore, the activity of the various cyclin–Cdk complexes results to be controlled by the synthesis of the appropriate cyclins during each specific phase of the cell cycle.

Cell Cycle Progression Inhibition in G1/S

Episodes of DNA damage during G1 pose a particular challenge because replication of damaged DNA can be deleterious and because no other chromatid is present to provide a template for recombinational repair. Besides, by considering that cyclins operate as promoting factors for mitosis and that typical cancer evolutions act as suppressors of certain members of the cyclins family, in case of DNA damage, the desired (healthy) state is identified by the G0 phase. Hence, in this context, we are interested to understand where and why G0 is reached.

5.6.1.1 p53-Dependent Checkpoint Pathway There are several proteins that can inhibit the cell cycle in G1 but, whenever a DNA damage occurs, p53[3] is the protein that gets accumulated in the cell and that induces the CyclinE_cdk2 p21-mediated inhibition. It can be activated by different proteins that, in turn, can be activated by different genotoxic or nongenotoxic stimuli. The role of this transcription factor is to induce the transcription of genes that encode proteins involved in apoptosis, of genes that encode proteins in charge to stop the cell cycle, and of proteins involved in the DNA repair machinery. When a damage is detected, p53 allows a cell a unique possibility for survival by starting the repair machinery. If this process fails, the cell is destined to die. In particular, whenever the DNA double strand is broken, p53 is activated by the ATM protein kinase. The oncoprotein Mdm2[4] binds the transcription

factor and blocks its activity through a dual mechanism: It conceals the p53 transactivation domain and promotes the p53 degradation after ubiquitination[5] [22]. ATM activates p53 preventing the Mdm2 binding, so its inhibitory effect cannot occur. This action allows p53 to shuttle to the nucleus. Here, it can promote the transcription of different target genes; one of them is a cyclin-dependent kinase inhibitor: p21. p21 is in charge to suppress the CyclinE_Cdk2 kinase activity, thereby resulting in G1 arrest [23].

This mechanism has been formalized using membrane systems and simulated in Mazza and Nocera [24]. In particular, we have extended the corresponding Reactome[6] [25] model (written in the Systems Biology Markup Language, SBML [26]). Moreover, we have translated the model into the membrane system framework [27] and have simulated its dynamics. The obtained membrane system model is described in Figure 5.6.

In addition to the described pathway, we have provided some extra rules with the aim to reduce any possible pathways cross-talk effects (in fact, very often, chemicals are involved in more than one living function and hence, they are involved in different pathways). Moreover, we have added an interaction rate to the rules, as described in Sedwards and Mazza [28], and we have used Cyto-Sim[7] to simulate the model.

We have initially employed the same quantitative initial configurations (except for Cyclin_Cdk2, which we set one-tenth of the others with the aim both to accelerate the degradation of p21 and to better qualitatively depict the arrest process) and same rate constants (except for the last two degradations and for the p21 binding, merely for complying qualitatively the well-known behaviors of the chemicals under examination).

As already mentioned before, we have added to the model some extra feedback rules in order to avoid pathways cross-talks issues. In particular, we have added a fictitious rule (r_9) that causes the consumption of the sequestered complex CyclinE_Cdk2 by p21 (r_8). In this way, we can monitor and temporize the cycle arrest process. Moreover, because damage, ATM, p53, and Mdm2 undergo phosphorylation and the corresponding ATMphospho, p53phospho, and Mdm2phospho are endlessly created, we have introduced three simple degradation rules (r_{10-12}) to take into account their balancing processes (that are, possibly, envisaged by other

pathways). When the modeled pathway is not perturbed by a DNA damage, the Mdm2_p53 complex is rapidly created (r_3) and quickly shuttled to cytoplasm (r_1), where it is degradated (r_{12}) (Figure 5.7). But when a damage occurs (r_2), the accumulation of Mdm2_p53 into the nucleus is quickly blocked (reducing its shuttling) and the accumulation of p53phospho is promptly triggered (r_6). After the damage, the quantity of Mdm2_p53 shuttled decreases (from 270 to 370 complexes), and the accumulated p53phospho molecules transcriptionally activate p21 (r_7) that accumulates and sequesters CyclinE_Cdk2 (r_8) for G1/S arrest (Figure 5.8).

$\Pi = (O, \mu, w_c, w_n, (u_c, v_c), (u_n, v_n), R^m, R)$, where

$O = \{damage, ATMdimer, ATMphospho, Mdm2, Mdm2\_p53, Mdm2phospho, p53phospho, p21, CyclinE\_Cdk2, p21\_CyclinE\_Cdk2\}$,

$\mu = [_c[_n]_n]_c$,

$w_c = \lambda$,

$w_n = damage^{1000}\ ATM\_dimer^{1000}\ Mdm2^{1000}\ p53^{1000}\ Cyclin\_Cdk2^{1000}$,

$u_c = \lambda; v_c = \lambda; u_n = \lambda; v_n = \lambda$,

$R^m = \{r_1: [Mdm2\_p53]^n \rightarrow [\ ]^n Mdm2\_p53\}$ $\quad rate(r_1) = 1$,

$R =$

$\{$

$r_2 : [damage + ATMdimer \rightarrow ATMphospho^2]^n$ $\quad rate(r_2) = 1$,

$r_3 : [Mdm2 + p53 \rightarrow Mdm2\_p53]^n$ $\quad rate(r_3) = 1$,

$r_4 : [Mdm2\_p53 \rightarrow \lambda]^c$ $\quad rate(r_4) = 1$,

$r_5 : [ATMphospho + Mdm2 \rightarrow ATMphospho + Mdm2phospho]^n$ $\quad rate(r_5) = 1$,

$r_6 : [ATMphospho + p53 \rightarrow ATMphospho + p53phospho]^n$ $\quad rate(r_6) = 1$,

$r_7 : [p53phospho \rightarrow p53phospho + p21]^n$ $\quad rate(r_7) = 1$,

$r_8 : [p21 + CyclinE\_Cdk2 \rightarrow p21\_CyclinE\_Cdk2]^n$ $\quad rate(r_8) = 0.8$,

$r_9 : [p21\_CyclinE\_Cdk2 \rightarrow \lambda]^n$ $\quad rate(r_9) = 1$,

$r_{10} : [ATMphospho \rightarrow \lambda]^n$ $\quad rate(r_{10}) = 1$,

$r_{11} : [p53phospho \rightarrow \lambda]^n$ $\quad rate(r_{11}) = 0.6$,

$r_{12} : [Mdm2phospho \rightarrow \lambda]^n$ $\quad rate(r_{12}) = 0.6$

$\}$

Figure 5.6 p53-dependent G1/S arrest. The membrane system is written in the style described in Section 5.5. However, with the aim to be closer to biochemistry, we use the symbol "+" to represent multiset concatenation (instead of just writing them by concatenating the symbols, as is usually done in the membrane systems area and as presented in Section 5.3). For instance, here a rule $[u_1 u_2 \rightarrow v_1 v_2]^1$ is written as $[u_1 + u_2 \rightarrow v_1 + v_2]^1$. Moreover, the labels used are short notations for the following cellular compartments: s = system, c = cytoplasm, and n = nucleoplasm.

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

SCHOOL OF BIO AND CHEMICAL ENGINEERING

DEPARTMENT OF BIOINFORMATICS

UNIT – III- SBIA5301 – SYSTEMS BIOLOGY

UNIT III BIOLOGICAL NETWORK INFERENCE Reconstruction of Biological Networks by Supervised Machine Learning Approaches - Graph Reconstruction as a Pattern Recognition Problem - Problem Formalization - Graph Inference as a Pattern Recognition Problem - Graph Inference with Local Models and Global Models - Examples - Reconstruction of a Metabolic Network - Reconstruction of a PPI Network Reconstruction of Gene Regulatory Networks

RECONSTRUCTION OF BIOLOGICAL NETWORKS BY SUPERVISED MACHINE LEARNING APPROACHES

In this review chapter, we focus on the problem of reconstructing the structure of large-scale biological networks. By biological networks, we mean graphs whose vertices are all or a subset of the genes and proteins encoded in a given organism of interest, and whose edges, either directed or undirected, represent various biological properties. As running examples, we consider the three following graphs, although the methods presented below may be applied to other biological networks as well.

- *Protein–protein interaction (PPI) network*. This is an undirected graph with no self-loop, which contains all proteins encoded by an organism as vertices. Two proteins are connected by an edge if they can physically interact.
- *Gene regulatory network*. This is a directed graph that contains all genes of an organism as vertices. Among the genes, some called transcription factors (TFs) regulate the expression of other genes through binding to the DNA. The edges of the graph connect TFs to the genes they regulate. Self-loops are possible

if a TF regulates itself. Moreover, each edge may in principle be labeled to indicate whether the regulation is a positive (activation) or negative (inhibition) regulation.

- *Metabolic network*. This graph contains only a subset of the genes as vertices, namely, those coding for enzymes. Enzymes are proteins whose main function is to catalyze a chemical reaction, transforming substrate molecules into product molecules. Two enzymes are connected in this graph if they can catalyze two successive reactions in a metabolic pathway, that is, two reactions, such that the main product of the first one is a substrate of the second one.

Deciphering these networks for model organisms, pathogens, or human is currently a major challenge in systems biology, with many expected applications ranging from basic biology to medical applications. For example, knowing the detailed interactions possible between proteins on a genomic scale would highlight key proteins that interact with many partners, which could be interesting drug targets [1], and would help in the annotation of proteins by annotation transfer between interacting proteins. The elucidation of gene regulatory networks, especially in bacteria and simple eukaryotes, would provide new insights into the complex mechanisms that allow an organism to regulate its metabolism and adapt itself to environmental changes and could provide interesting guidelines for the design of new functions. Finally, understanding, in detail, the metabolism of an organism and clarifying which proteins are in charge of its control, would give a valuable description of how organisms have found original pathways for degradation and synthesis of various molecules, and could help again in the identification of new drug targets [2].

Decades of research in molecular biology and genetics have already provided a partial view of these networks, in particular, for model organisms. Moreover, recent high-throughput technologies such as the yeast two-hybrid systems for PPI provide large numbers of likely edges in these graphs, although probably with a high rate of false positives [3, 4]. Thus, much work remains to be done in order to complete (adding currently unknown edges) and correct (removing false-positive edges) these partially known networks. To do so, one may want to use information about individual genes and proteins such as their sequence, structure, subcellular localization, or level of expression across several experiments. Indeed, this information often provides useful hints about the presence or absence of edges between two proteins. For example, two proteins are more likely to interact physically if they are expressed in similar experiments and localized in the same cellular compartment, or two enzymes are more likely to be involved in the same metabolic pathway if they are often coexpressed and if they have homologs in the same species [5–7].

Following this line of thought, many approaches have been proposed in the recent years to infer biological networks from genomic and proteomic data, most of them attempting to reconstruct the graphs *de novo*. In *de novo* inference, the data about individual genes and proteins are given and edges are inferred from these data only, using a variety of inference principles. For example, when time series of expression data are used, regulatory networks have been reconstructed by fitting various dynamical system equations to the data [8–14]. Bayesian networks have also been used to

infer *de novo* regulatory networks from expression data, assuming that direct regulation can be inferred from the analysis of correlation and conditional independence between expression levels [15]. Another rationale for *de novo* inference is to connect genes or proteins that are similar to each other in some sense [5, 6]. For example, coexpression networks or the detection of similar phylogenetic profiles are popular ways to infer "functional relationships" between proteins, although the meaning of the resulting edges has no clear biological justification [16]. Similarly, some authors have attempted to predict gene regulatory networks by detecting large mutual information between expression levels of a TF and the genes it regulates [17, 18].

In contrast to these *de novo* methods, in this review, we present a general approach to reconstruct biological networks using information about individual genes and proteins based on supervised machine learning algorithms, as developed through a recent series of articles [19–26]. The graph inference paradigm we follow assumes that, besides the information about individual vertices (genes or proteins) used by *de novo* approaches, the graph we wish to infer is also partially known, and known edges can be used by the inference algorithm to infer unknown edges. This paradigm is similar to the notion of supervised inference in statistics and machine learning, where one uses a set of input/output pairs (often called the training set) to estimate a function that can predict the output associated with new inputs [27, 28]. In our paradigm, we give us the right to use the known edges of the graph to supervise the estimation of a function that could predict whether a new pair of vertices is connected by an edge or not, given the data about the vertices. Intuitively, this setting can allow us to automatically learn what features of the data about vertices are the most informative to predict the presence of an edge between two vertices. In a sense, this paradigm leads to a problem much simpler than the *de novo* inference problem, since more information is used as an input, and it might seem unfair to compare *de novo* and supervised methods. However, as already mentioned, in many real-world cases of interest, we already partially know the graph we wish to infer. It is, therefore, quite natural to use as much information as we can in order to focus on the real problem, which is to infer new edges (and perhaps delete wrong edges), and, therefore, to use as an input both the genomic and proteomic data, on the one hand, and the edges already known, on the other.

In a slightly more formal language, we, therefore, wish to learn a function that can predict whether an edge exists or not between two vertices (genes or proteins), given data about the vertices (e.g., expression levels of each gene in different experimental conditions). Technically, this problem can be thought of as a problem of binary classification, where we need to assign a binary label (presence or absence of an edge) to each pair of vertices, as explained in Section 7.2.1. From a computational point of view, the supervised inference paradigm we investigate can, in principle, benefit from the availability of a number of methods for supervised binary classification, also known as pattern recognition [28]. These methods, as reviewed in Section 7.2.2, are able to estimate a function to predict a binary label from data about patterns, given a training set of (pattern, label) pairs. The supervised inference problem we are confronted with, however, is not a classical pattern/label problem because the data are associated with individual vertices (e.g., expression profiles are available for each individual gene), while the labels correspond to pairs of vertices. Before applying out-of-the-box state-of-the-art machine learning algorithms, we, therefore, need to clarify how our problem can be transformed as a classical pattern recognition problem (Section 7.2.3). In particular, we show that there is not a unique way to do that, and present in Sections 7.2.4 and 7.2.5, two classes of approaches that have been proposed recently. Both classes involve a support vector machine (SVM) as a binary classification engine, but follow different avenues to cast the edge inference problem as a binary classification problem. In Section 7.3, we provide experimental results that justify the relevance of supervised inference and show that a particular approach, based on local models, performs particularly well on the reconstruction of PPI and regulatory and metabolic networks. We conclude with a rapid discussion in Section 7.4.

GRAPH RECONSTRUCTION AS A PATTERN RECOGNITION PROBLEM

In this section, we formally define the graph reconstruction problem considered and explain how to solve it with pattern recognition techniques.

We consider a finite set of vertices $V = (v_1, \ldots, v_n)$ that typically correspond to the set of all genes or proteins of an organism. We further assume that for each vertex $v \in V$, we have a description of various features of v as a vector $\phi(v) \in \mathbb{R}^p$. Typically, $\phi(v)$ could be a vector of expression levels of the gene v in p different experimental conditions, measured by DNA microarrays, a phylogenetic profile that encodes the presence or absence of the gene in a set of p sequenced genomes [6], a vector of p sequence features, or a combination of such features. We wish to reconstruct a set of edges $E \subset V \times V$ that defines a biological network. While in *de novo* inference, the goal is to design an algorithm that automatically predicts edges in E from the set of vertex features $(\phi(v_1), \ldots, \phi(v_n))$, in our approach, we further assume that a set of pairs of vertices known to be connected by an edge or not is given. In other words, we assume given a list $\mathcal{S} = ((e_1, y_1), \ldots, (e_N, y_N))$ of pairs of vertices $(e_i \in V \times V)$ tagged with a label $y_i \in \{-1, 1\}$ that indicate whether the pair e_i is known to interact

($y_i = 1$) or not ($y_i = -1$). In an ideal noise-free situation, where the labels of pairs in the training set are known with certainty, we thus have $y_i = 1$ if $e_i \in E$, and $y_i = -1$ otherwise. However, in some situations, we may also have noise or errors in the training set labels, in which case, we could only assume that pairs in E tend to have a positive label, while pairs not in E tend to have a negative label.

The graph reconstruction problem can now be formally stated as follows: Given the training set $\mathcal{S}$ and the set of vertex features $(\phi(v_1), \ldots, \phi(v_n))$, predict for all pairs not in $\mathcal{S}$ whether they interact (i.e., whether they are in E) or not. This formulation is illustrated in Figure 7.1.

Stated this way, this problem is similar to a classical pattern recognition problem, for which a variety of efficient algorithms have been developed over the years. Before highlighting the slight difference between the classical pattern recognition framework

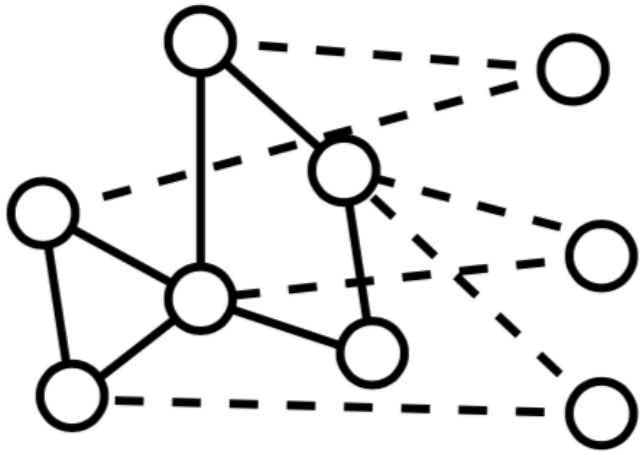

Figure 7.1 We consider the problem of inferring missing edges in a graph (dotted edges), where a few edges are already known (solid edges). To carry out the inference, we use attributes available about individual vertices such as vectors of expression levels across different experiments if vertices are genes.

and ours, it is, therefore, worth recalling this classical pattern recognition paradigm and mentioning some algorithms adapted to solve it.

Pattern Recognition

Pattern recognition, of binary supervised classification, is a well-studied problem in statistics and machine learning [27, 28]. In its basic setup, a training set $\mathcal{T} = \{(u_1, t_1), \ldots, (u_N, t_N)\}$ of labeled patterns is given, where $u_i \in \mathbb{R}^q$ is a vector and $t_i \in \{-1, 1\}$ is a binary label, for $i = 1, \ldots, N$. The goal is then to infer a function $f : \mathbb{R}^q \to \{-1, 1\}$ that is able to predict the binary label t of any new pattern $u \in \mathbb{R}^q$ by $f(u)$.

Many methods have been proposed to infer the labeling function f from the training set $\mathcal{T}$, including, for example, nearest neighbor classifiers, decision trees, logistic regression, artificial neural networks, or SVMs. Although any of these methods can be used in what follows, we will present experiments carried out with an SVM, which we briefly describe below, mainly for three reasons:

- It is now a widely used algorithm, in particular, in computational biology, with many public implementations [29, 30].
- It provides a convenient framework to combine heterogeneous features about the vertices such as the sequence, expression, and subcellular localization of proteins [19, 31, 32].
- Some methods developed so far for graph inference, which we describe below, are particularly well-adapted for a formalization in the context of SVM and kernel methods [22, 24].

Let us, therefore, briefly describe the SVM algorithm and redirect the interested reader to various textbooks for more details [33–35]. Given the labeled training set $\mathcal{T}$, an

SVM estimates a linear function $h(u) = w^\top u$ for some vector $w \in \mathbb{R}^q$ (here $w^\top u$ represents the inner product between w and u) and then makes a label prediction for a new pattern u that depends only on the sign of $h(u)$: $f(u) = 1$ if $h(u) \geq 0$, $f(u) = -1$ otherwise. The vector w is obtained as the solution of an optimization problem that attempts to enforce a correct sign with large absolute values for the values $h(u_i)$ on the training set while controlling the Euclidean norm of w. The resulting optimization problem is a quadratic program for which many specific and fast implementations have been proposed.

Graph Inference as a Pattern Recognition Problem

Let us now return to the graph reconstruction problem, as presented in Section 7.2.1. At first sight, this problem is very similar to the general pattern recognition paradigm recalled in Section 7.2.2: Given pairs of vertices with positive and negative labels, infer a function f to predict whether a new pair has a positive label (i.e., is connected) or not. An important difference between the two problems, however, is that the features available in the graph reconstruction problem describe properties of individual vertices v and not of pairs of vertices (v, v'). Thus, in order to apply pattern recognition techniques such as the SVM to solve the graph reconstruction problem, we can follow one of the two possible avenues.

(1) Reformulate the graph reconstruction problem as a pattern recognition problem, where binary labels are attached to individual vertices (and not to pairs of vertices). Then pattern recognition methods can be used to infer the label of vertices based on their features.
(2) Keep the formulation as the problem of predicting the binary label of a pair of vertices, but find a way to represent as vectors (or as a kernel) pairs of vertices, while we initially only have features for individual vertices.

Graph Inference with Local Models

In this section, we describe an approach that was proposed by Bleakley et al. [25] for the reconstruction of metabolic and PPI networks and successfully applied by Mordelet and Vert [26] for regulatory network inference. The basic idea is very simple and can be thought of as a "divide-and-conquer" strategy to infer new edges in a graph. Each vertex of the graph is considered in turn as a seed vertex, independently from the others, and a "local" pattern recognition problem is solved to discriminate the vertices that are connected to this seed vertex against the vertices that are not connected to it. The local model can then be applied to predict new edges between the seed vertex and other vertices. This process is then repeated with other vertices as seed to obtain edge prediction throughout the graph. More precisely, the "local model" approach can be described as follows:

(1) Take a seed vertex v_{seed} in V.
(2) For each pair (v_{seed}, v') with label y in the training set, associate the same label y with the individual vertex v'. This results in a set of labeled vertices $\left\{(v'_1, t_1), \ldots, (v'_{n(v_{\text{seed}})}, t_{n(v_{\text{seed}})})\right\}$, where $n(v_{\text{seed}})$ is the number of pairs starting with v_{seed} in the training set. We call this set a local training set.
(3) Train a pattern recognition algorithm on the local training set designed in step 2.
(4) Predict the label of any vertex v' that has no label, that is, such that (v_{seed}, v') is not in the training set.
(5) If a vertex v' has a positive predicted label, then predict that the pair (v_{seed}, v') has a positive label (i.e., is an edge).
(6) Repeat steps (1)–(5) for each vertex v_{seed} in V.
(7) Combine the edges predicted at each iteration together to obtain the final list of predicted edges.

Reconstruction of a Metabolic Network

The reconstruction of metabolic networks has been among the first applications that motivated the line of research surveyed in this Chapter [19–21, 25]. We consider here the problem of inferring the metabolic gene network of the yeast *S. cerevisiae* with the enzymes represented as vertices, and an edge between two enzymes when the two enzymes catalyze successive reactions. The dataset, proposed by Yamanishi et al. [21], consists of 668 vertices (enzymes) and 2782 edges between them, which were extracted from the KEGG database of metabolic pathways [38]. In order to predict edges in these networks, Bleakley et al. [25] used various genomic datasets and compared different inference methods. Following Yamanishi et al. [21], the data used to characterize enzymes comprise 157 expression data measured under different experimental conditions [39, 40], a vector of 23 bits representing the localization of the enzymes (found or not found) in 23 locations in the cell determined experimentally [41], and the phylogenetic profiles of the enzymes as vectors of 145 bits denoting the presence or absence of the enzyme in 145 fully sequenced genomes [38]. Each type of data was processed and transformed into a kernel as described in Yamanishi et al. and Kato et al. [21, 42], and all matrices were summed together to produce a single kernel integrating heterogeneous data.

On a common five-fold cross-validation setting, Bleakley et al. [25] compared different methods including local models (Section 7.2.4), the TPPK and MLPK kernels (Section 7.2.5) as well as several other methods: a direct *de novo* approach, which only infers edges between similar vertices, an approach based on kernel canonical correlation analysis (KCCA) [19], and a matrix completion algorithm based on an *em* procedure [42, 43]. On each fold of the cross-validation procedure, each method uses the training set to learn a model and makes predictions on pairs in the test set. All methods associate a score with all pairs in the test set, hence by thresholding this score at different levels, they can predict more or less edges. Results were assessed in terms of average ROC curve (which plots the percentage of true positives as a function of the percentage of false positives, when the threshold level is varied) and average precision/recall curve (which plots the percentage of true positives among positive predictions, as a function of the percentage of true positives among all positives). In practical applications, the later criterion is a better indicator of the relevance of a method than the former one. Indeed, as biological networks are usually sparse, the number of negatives far exceeds the number of positives, and only large precision (over a recall as large as possible) can be tolerated if further experimental validations are expected.

Figure 7.4 shows the performance of the different methods on this benchmark. A very clear advantage for the local model can be seen. In particular, it is the only method tested that can produce predictions at more than 80 percent precision. There is no clear winner among the other supervised methods, while the direct approach, which is the only *de novo* method in this comparison, is clearly below the supervised methods.

Reconstruction of a PPI Network

As a second application, we consider the problem of inferring missing edges in the PPI network of the yeast *S. cerevisiae*. The gold standard PPI graph used to perform a cross-validation experiment is a set of high-confidence interactions supported by several experiments provided by Von Mering et al. [44] and also used in Kato et al. [42]. After removal of proteins without interactions, we end up with a graph involving 2438 interactions (edges) among 984 proteins (vertices). In order to reconstruct missing edges, the genomic data used are the same as those used for the reconstruction of the metabolic network in Section 7.3.1, namely, gene expression, protein localization, and phylogenetic profiles, together with a set of yeast two-hybrid data obtained from Uetz et al. [3] and Ito et al. [4]. The later was converted into a positive definite kernel using a diffusion kernel, as explained in Kato et al. [42]. Again, all datasets were combined into a unique kernel by adding together the four individual kernels.

Figure 7.5 shows the performances of the different methods, using the same experimental protocol as the one used for the experiment with metabolic network reconstruction in Section 7.3.1. Again, the best method is the local model, although it outperforms the other methods with a smaller margin than for the reconstruction of the metabolic network (Figure 7.4). Again, the ROC curve of the *de novo* direct method is clearly below the curves of the supervised methods, although this time it leads to a large precision at low recall. This means that a few interacting pairs can very easily be detected because they have very similar genomic data.

Reconstruction of Gene Regulatory Networks

Finally, we report the results of an experiment conducted for the inference of a gene regulatory network by Mordelet and Vert [26]. In that case, the edges between transcription factors and the genes they regulate are directed; therefore, only the local model of Section 7.2.4 is tested. It is compared with a panel of other state-of-the-art methods dedicated to the inference of gene regulatory networks from a compendium of gene expression data, using a benchmark proposed by Faith et al. [18]. More precisely, the goal of this experiment is to predict the regulatory network of the bacteria *Escherichia coli* from a compendium of 445 microarray expression profiles for 4345 genes. The microarray was collected under different experimental conditions such as pH changes, growth phases, antibiotics, heat shock, different media, varying oxygen concentrations, and numerous genetic perturbations. The goal standard graph used to assess the performance of different methods by cross-validation consists of 3293 experimentally confirmed regulations between 154 TF and 1211 genes, extracted from the RegulonDB database [45].

In Faith et al. [18], this benchmark was used to compare different algorithms, including Bayesian networks [15], ARACNe [46], and the context likelihood of relatedness (CLR) algorithm [18], a new method that extends the relevance networks class of algorithms [17]. They observed that CLR outperformed all other methods in prediction accuracy and experimentally validated some predictions. CLR can, therefore, be considered as the state-of-the-art among methods that use compendia of gene expression data for large-scale inference of regulatory networks. However, all the methods compared in Faith et al. [18] are *de novo*, and the goal of Mordelet and Vert [26] was to compare the supervised local approach to the best *de novo* method on this benchmark, namely, the CLR algorithm. Using a three-fold cross-validation procedure (see details in Mordelet and Vert [26]), they obtained the curves in Figure 7.6. We can observe that the local supervised approach (called SIRENE for Supervised Inference of REgulatory NEtwork) strongly outperforms the CLR method on this benchmark. The recall obtained by SIRENE, that is, the proportion of known regulations that are correctly predicted, is several times larger than the recall of CLR at all levels of precision. More precisely, Table 7.1 compares the recalls of SIRENE, CLR, and several other methods at 80 percent and 60 percent precision. The other methods reported are relevance network [17], ARACNe [46], and a Bayesian network [15] implemented by Faith et al. [18].

Table 7.1 Recall of different gene regulation prediction algorithms at different levels of precision (60% and 80%)

Method	Recall at 60%	Recall at 80%
SIRENE	44.5%	17.6%
CLR	7.5%	5.5%
Relevance networks	4.7%	3.3%
ARACNe	1%	0%
Bayesian network	1%	0%

Source: From Ref. 26.

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

SCHOOL OF BIO AND CHEMICAL ENGINEERING

DEPARTMENT OF BIOINFORMATICS

UNIT – IV- SBIA5301 – SYSTEMS BIOLOGY

FROM DNA MOTIFS TO GENE NETWORKS: A REVIEW OF PHYSICAL INTERACTION MODELS

Understanding the interactions between biomolecules within a cell and between cells and their environment is one of the major challenges in computational biology. Although every cell in an organism contains the same genetic material, its expression profile depends on the tissue type, developmental stage, and the extracellular signals it receives at the given point in time. Cells exert various ways to regulate the expression of their genes. Chromatin structure, for example, can make large parts of the genome transcriptionally silent or potentially active. Also, posttranscriptional and posttranslational mechanisms can influence the amount and the activity of the available proteins and noncoding genes in a cell. The best studied mechanism for gene expression control, however, is the transcription regulation at the individual

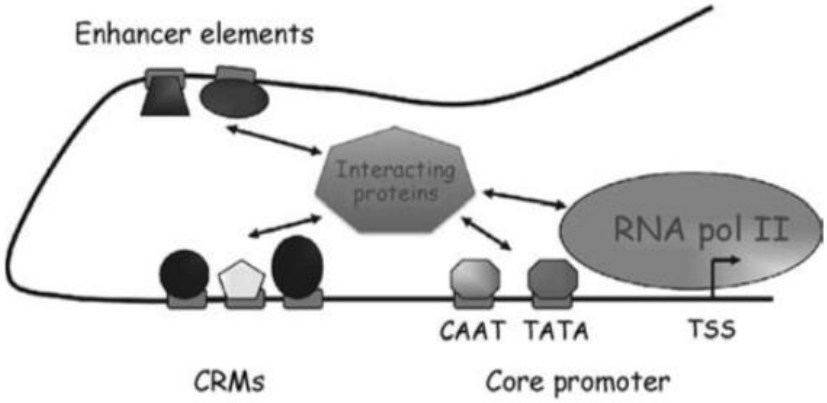

Figure 12.1 Schematic representation of a typical eukaryotic promoter. The transcription start site (TSS), the core promoter, and the enhancer elements are depicted.

gene level. Transcription factor (TF) proteins recognize short DNA "signals" (typically 6–15 base pairs long) in the vicinity of the genes' transcription start sites (TSSs) and enhance or suppress their expression. These DNA signals are commonly referred to as transcription-factor binding sites (TFBSs) or—more general—as *cis*-regulatory elements. A broad classification of the role of these regulatory elements can be done on the basis of their distance from the gene's TSS (Figure 12.1). The region located in the first 300–500-bp upstream of the TSS constitutes the core promoter of the gene and frequently contains binding sites for general TFs, like the TATA-box and the CAAT-box. Core promoters are relatively conserved regions across all vertebrates [1, 2]. Farther upstream are located the TF binding sites that are responsible for the

gene's expression specificity (i.e., when and where the gene is expressed). The timely and tissue-specific expression of all genes is crucial for the cell itself and the organism as a whole. Expression is usually regulated by sets of TFs, whose binding sites are closely located in the genome, and the TFs themselves can directly interact with each other. These sets of sites are known as *cis*-regulatory modules (CRMs.) CRMs can be found few kilobases around the TSS. Finally, in complex eukaryotic organisms, some TF target sequences can be found tens of thousands of bases away from the TSS. These regions are usually called enhancers, and their main role is to fine-tune genes' regulation, usually through protein–protein interactions. Figure 12.1 presents some of the features of a typical eukaryotic promoter.

TF genes interact with each other either directly (i.e., by forming protein complexes) or indirectly (i.e., by regulating each other's expression). TFs act individually (as monomers) or in complexes (as homo- or hetero-multimers). This creates a network of interactions that characterizes a cell's response to a particular stimulus. Focusing only on TF genes, one can construct the network of all regulatory interactions. Figure 12.2 shows some of the simple components of the TF interaction networks that have been observed [3]. Reverse engineering refers to the traditional mathematical inverse problem, which is to infer the gene regulatory circuit (network topology) from gene expression data. Genes can be represented as nodes in a graph, where edges represent the direct interactions between genes. There are two broad classes of reverse-engineering algorithms for gene regulatory networks [4]: those based on

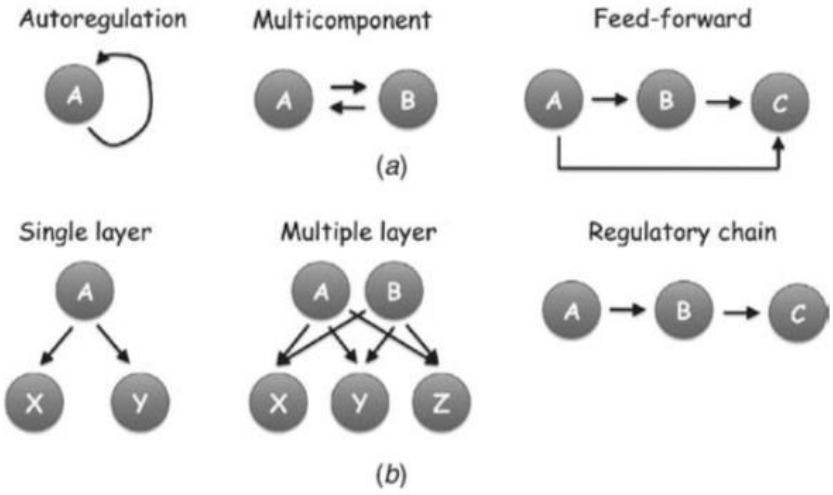

Figure 12.2 Network components. Regulatory network components are presented with respect to their interactions (a) or the layers of the signal transduction (b). Data from Alon [3] and Lee et al. [48].

the "physical interactions" that aim at identifying interactions among transcription factors and their target genes (gene-to-sequence) and those based on the "influence interactions" that try to relate the expression of a gene to the expression of the other genes in the cell (gene-to-gene).

A number of approaches have been developed for modeling regulatory networks. A broad taxonomical organization suggests four major methodological categories for these approaches. The first includes optimization methods based on the maximization of a high-dimensional objective function associated with different network topologies such as Bayesian networks [5, 6] or chain functions [7]. An objective function used frequently is the log-likelihood of the network topology given the observed data. The second category includes a variety of regression techniques to fit the observed data to an empirical *a priori* model of the underlying biochemical interactions [8–10]. A third group includes integrative bioinformatics approaches that combine data from a number of independent clues, such as known protein–protein and protein–DNA interactions (from databases or literature), expression data, or DNA binding motifs [11–13]. The fourth category includes statistical/information theoretical methods [14, 15], which define two-way or higher order probabilistic measures of gene correlation to distinguish potential interactions from background noise. Models of gene regulatory networks can also be divided according to the representation of the network states (discrete vs. continuous), the nature of the data (static vs. dynamic over time or different conditions), the representation of gene associations (qualitative vs. quantitative), the dependencies between genes (linear vs. nonlinear), the nature of the model (deterministic vs. stochastic), and the location of the genes in the cells (nonspatial vs. spatial).

This chapter focuses on the "physical interaction" networks. First, we will give an overview of the physical basis of transcription regulation and the representation of the regulatory DNA patterns. Then, we will survey some of the physical interaction algorithms for reverse engineering of gene expression data. The coverage of the algorithms is not exhaustive and is biased toward what we believe are the more practical methods. We attempt to cover at least one method from each class of algorithms of this broad category.

Physical Basis of Transcription Regulation and Representation of DNA Patterns

Each TF recognizes a set of DNA binding sites with high affinity. It usually achieves this by placing one or more α-helices in the major groove of the DNA. The specific DNA target recognition results from the molecular contacts (hydrogen bonds, electrostatic interactions, etc.) between the amino acids and the DNA bases. Contacts from and to the backbone of the protein or DNA also contribute to the overall binding affinity (how strongly a target sequence is bound), although their contribution to binding specificity (how more strongly a sequence is bound compared to a random sequence) is generally assumed to be secondary [16]. Sometimes, nonbase-specific DNA interactions contribute to the target recognition. This is usually referred in the literature as "indirect readout." An example is the *CAP* (or *CPR*) protein, which bends the DNA upon binding. In this case, in addition to the specific base–amino acid contacts, the overall sequence of the DNA target needs to have some degree of "bendability," thus restricting further the repertoire of tolerated changes.

Preferred binding sites of a TF can be discovered and verified by *in vitro* target selection experiments (e.g., SELEX [17] or protein-binding microarrays [18]) or by biochemical analysis of the upstream regions of its known target genes. The length and the number of optimal targets vary, depending on the TF in question. For example, *c-myc* oncogene in mammals and Ultrabiothorax (*Ubx*) gene in *Drosophila* have a very restricted set of targets (CACGTG and ATTA, respectively), whereas the pattern of p53 is more degenerate (Figure 12.3). There are many ways to represent the TF binding preferences [19], but the most popular so far has been proven to be the position-specific scoring matrices (PSSMs) or position weight matrices (PWMs.)

PSSM models are 4X*L* weight matrices, where *L* is the length of the DNA binding motif (the single targets of most TFs are of a given length *L*, which is a characteristic of the TF). To generate a PSSM model, the known sites of a given TF are aligned and a *4XL* frequency table is calculated. Column *I* in this table consists of the four base frequencies at position *I* of the alignment. The PSSM model typically consists of the log-likelihood ratios of the observed frequencies against the background frequency of the corresponding base. We note that the average log-likelihood ratio in each position is the relative entropy, formally defined as:

$$\mathrm{RH}(I) = \sum_{b=A}^{T} f(b, I) \quad \ln \frac{f(b, I)}{P_{\mathrm{ref}}(b)}, \tag{12.1}$$

where $f(b,I)$ is the estimated frequency of base b at position I of the pattern and $P_{\mathrm{ref}}(b)$ is background frequency of base b (e.g., in the genome). Averaging over all

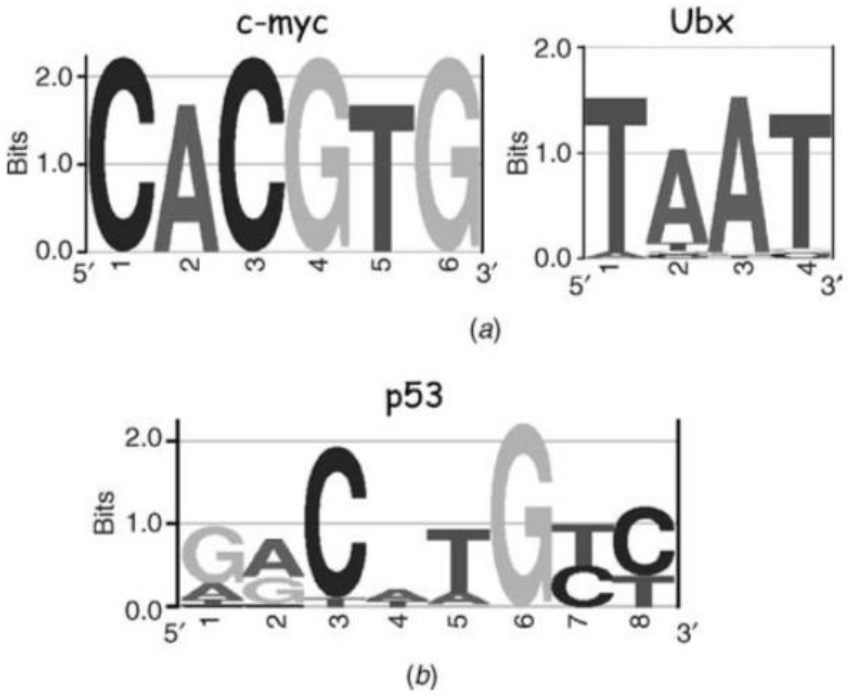

Figure 12.3 Types of DNA patterns. *Ubx* and *c-myc* have very restricted repertoire of binding sites (a), whereas *p53* targets are more degenerate (b).

L positions, we obtain the average relative entropy of the motif. A number of motif finding algorithms identify patterns that maximize either the overall log-likelihood of the motif or its relative entropy.

There is an interesting theoretical interpretation of the PSSM models. Through a Boltzmann theory perspective, a PSSM model can be viewed as the average binding specificity of a TF protein to its DNA targets. There are examples in the literature that show that the PSSM score is in agreement with binding energy measurements [20]. One general assumption of the PSSM models is the position independence, that is, the observed base frequencies in one position are independent of the frequencies in any other position. According to the thermodynamic model, this corresponds to energetic additivity, that is, each base position contributes independently of the others to the total binding energy. Energetic additivity is a simplification of the physical properties of the TF–DNA interactions and it does not hold in general [21, 22]. However, in practice, it has been found to be a good approximation in modeling binding affinities [23]. In addition, additive models require a significantly smaller number of parameters. These two properties have made additive models very useful and contributed to their popularity.

PHYSICAL INTERACTION ALGORITHMS

Physical interaction algorithms are those reverse-engineering algorithms that aim at identifying interactions among TFs and their target genes (gene-to-sequence interactions). Microarray measurements, of course, do not necessarily reflect the transcription factor activities (TFAs) in all cases, since posttranscriptional and posttranslational modifications may play an important role for determining the activity of a TF. Nevertheless, for practical purposes, all physical network algorithms focus on the TFAs that can be deduced from microarray data.

Practically, physical interaction algorithms have two goals. One is to identify the genes regulated by a TF (or a set of TFs). In other words, they aim to reconstruct the connectivity structure and weights of the network. Second, they aim to reconstruct the activity profile of each TF from the gene expression data. An advantage of this strategy, compared to influence interaction algorithms, is that it reduces the dimensionality of the problem by analyzing only the interactions between TFs and their putative target genes (instead of all-against-all). It also enables the use of genome sequence data ("static data"), in combination with gene expression data ("dynamic data"), in order to enhance the sensitivity and specificity of the predicted interactions. The limitation of this approach is that it can only describe the regulatory control exercised by TFs.

Some physical interaction algorithms depend on the prediction of sets of coregulated genes, while others construct a more general model without such assumptions. Some of the algorithms represent regulatory activities as a function of the mRNA measurements, while others treat the regulatory activities as hidden variables. Finally, some of the algorithms can model complex *cis*-regulatory logic of multiple interacting TFs binding closely located DNA targets (*cis*-regulatory modules or CRMs). In the following, we will review four classes of physical interaction algorithms: the clustering-based approaches, the regression-based models, the network component analysis methods, and the factor analysis methods.

GENOMIC SIGNATURES AND A SPACE OF GENOMES FOR GENOME COMPARISON

On the basis of this analysis, we propose a novel formal framework to interpret genomic relationships derived from entire genome sequences rather than individual loci. This space allows to analyze sets of organisms related by a common *codon bias signature* (at times, more than one kind of bias influences the same genomic sequence and the ensemble of these overlapped biases defines what we call the *signature* of a genome) [34]. We give a number of numerical criteria to infer content bias, translational bias, and strand bias for genome sequences. We show in a uniform framework that genomes of quite different phylogenetic relationship share similar codon bias; other genomes grouped together by various phylogenetic methods appear to be subdivided into finer subgroups sharing different codon bias characteristics; Archaea and Eubacteria share the same codon preferences when $AT3$ or $GC3$ bias is their dominant bias; archaeal genomes satisfying translational bias use a more sharply distinguished set of preferred codons than bacterial genomes do. Our analysis, based on 96 eubacterial and archaeal genomes, opens the possibility that this space might reflect the geometry of a prokaryotic "physiology space". If this turns out to be the case,

the combination of the upcoming sequencing of entire genomes and the detection of codon bias signatures will become a valuable tool to infer information on the physiology, ecology, and possibly, ecological conditions under which bacterial and archaeal organisms evolved. For many organisms, this information would be impossible to be detected otherwise. More recently, our algorithm has been applied to more than 300 genomes and our hypothesis of environmental signature has been supported at larger scale [35].

Spaces for environmental and physiological classification represent a bacterial classification alternative to phylogeny and they are closer to the living conditions of the organism. With a growing number of genomic data available, it becomes more and more important to have new alternative organizational schemes to understand bacterial populations and the biology of single organisms within their living environment. The algorithmic idea working for bacteria should be revisited for metagenomic sequences for instance and adapted for viral genomes. On such spaces, hypotheses such as adaptability of a virus to the codon bias of its host can be checked and preliminary analysis support this hypothesis (see Section 14.8).

Genes with high codon bias describe in meaningful ways the biological characteristics of the organism and are representative of specific metabolic usage [36]. *In silico* methods exploiting this basic principle are expected to become important in learning about the lifestyle of an organism and explain its evolution in the wild. We demonstrate that besides high expressivity during fast growth or glycolytic activities, which have been very often reported, the necessity for survival under specific biological conditions has its traces in the genetic coding [36]. This observation opens the possibility to predict rare but necessary metabolic activities through genome analysis.

High expression of certain classes of genes, like those constituting the translational machinery or those involved in glycolysis, are correlated particularly well in the case of fast-growing organisms. By shifting the paradigm toward metabolic pathways, we notice that several energy metabolism pathways are correlated with high codon bias in organisms known to be driven by very different physiologies, which are not necessarily fast growing and whose genomes might be very homogeneous. More generally, we derive a classification of metabolic pathways induced by codon analysis and show that genetic coding for different organisms is tuned on specific pathways and that this is a universal fact. The codon composition of enzymes involved in glycolysis for instance, often required to be rapidly translated, is highly biased by dominant codon composition across species (this is indicated by the high CAI value of these enzymes). In fast growers, the numerical evidence is definitely far more striking than for other organisms (that is, the absolute difference between the CAI value of these enzymes and the average CAI value for genes in the genome is "large"), but even for *Helicobacter pylori*, a genome of rather homogeneous codon composition, enzymes involved in glycolytic pathways happen to be biased above average. In the same manner, one detects the crucial role of photosynthetic pathways for *Synechocystis* or of methane metabolism for *Methanobacterium*.

mRNA transcriptional levels collected during the *S. cerevisiae* cell cycle under diauxic shift [37] (here, glucose quantities decrease in the media during cell cycle and yeast goes from fermentation to aerobic respiration), have been used to analyze the yeast metabolic network in a similar spirit as done with codon analysis. A classification of metabolic pathways based on transcriptomic data has been proposed, and we show that the metabolic classification obtained through codon analysis essentially "coincides" with the one based on (a large and differentiated pool of) transcriptomic data. Such a result opens the way to explain evolutionary pressure and natural selection for organisms grown in the wild, and hopefully, to explain metabolism for slow-growing bacteria, as well as to suggest best conditions of growth in the laboratory.

It is an open question whether this kind of analysis can contribute to the reconstruction metabolic information from metagenomics data.

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

SCHOOL OF BIO AND CHEMICAL ENGINEERING

DEPARTMENT OF BIOINFORMATICS

UNIT – V- SBIA5301 – SYSTEMS BIOLOGY

ALI BABA: A TEXT MINING TOOL FOR SYSTEMS BIOLOGY

Text mining is the process of automatically deriving information from text (as opposed to data mining that works on structured data). This process starts with accessing the relevant literature and ends with extracting the desired pieces of information. Access mostly is provided by Web-based search tools, the best known of which is PubMed [1]. PubMed currently contains citations from close to 18 million publications in the biomedical domain (biology, biochemistry, medicine, and related fields), from approximately 5200 journals, since 1865. Up to 4000 citations (abstract and bibliographical information) are added to PubMed per day, which necessitates automated means to efficiently handle searches for high-quality information.

Text mining falls into several tasks, most of which depend on each other, but few of which have been sufficiently solved. The first task is information retrieval (IR): given a user's query, find the (most) relevant documents containing the keywords or, even better, providing an answer to the question the user actually has in mind. The later part is also called question answering (QA), where the task is not only to find relevant documents but also to extract the answer to the query from them. Information retrieval is often solved by keyword queries, as in PubMed [1], which returns the most recent abstracts containing the query. PubMed goes a step farther, expanding the initial set of keywords to related terms: a search for "cancer" will also find abstracts that mention neoplasms instead of cancer. Another related task is text summarization, which aims to summarize one or multiple documents with respect to a certain problem. An example is Entrez Gene [2], a database of genes, which contains a short summary for every entry, describing known functions, implications in diseases, and so on of the gene or the gene's products. These summaries are currently all manually compiled from various publications studying the gene and significant efforts are under way to automatize this curation process.

The next groups of tasks for text mining relates to information extraction; the most prominent is named entity recognition (NER), referring to the search for genes, proteins, diseases, drugs, and so on, mentioned in a text (we will call these biomedical entities in the remainder). In addition, instead of only recognizing that a name refers to a particular class of entities, entity mention normalization (EMN) tries to actually identify the entity, usually by searching a reference to a database. For instance, consider the name *p53*, which may stand for a large number of different yet orthologous genes; the task for EMN is to pick, when *p53* appears in a text, the correct one of these genes by identifying a corresponding database entry, for instance in Entrez Gene. Only then can the right set of additional information (function, species, sequence, etc.) become available to the user. Word sense disambiguation (WSD), on the other hand, tries to tell apart entities of different kinds that share the same name; *cancer* mostly refers to a disease, but in some contexts, it also refers to the genus of various crab species. Once entities are recognized, classified, and properly resolved, relation mining (RM) searches for evidence for associations between them, such as protein–protein interactions or gene–disease associations.

In the biological domain, an abundance of data of various types, degrees of detail, and quality is available. Much of these are stored in curated databases, that is, databases whose content is maintained by human experts. Among these databases, some store information on single types of biomedical objects, such as proteins (e.g., UniProt [3]), genes (e.g., NCBI Entrez Gene [2]), and drugs (e.g., DrugBank [4]); or on associations between these, such as protein–protein interactions (IntAct [5], MINT [6], etc.), drug–protein and target–disease relations (TTD [7] etc.), or metabolic pathways and other processes (e.g., KEGG [8]). The curation process for most of these databases relies on trained experts extracting supportive information from scientific publications and updating the database accordingly. Far from being able to deliver off-the-shelf solutions for handling such curation automatically, research in text mining currently focuses on aiding database curators and researchers in biology, medicine, and interdisciplinary fields, who search for single, specific, and accurate pieces of information in literature collections. With novel high-throughput data generation techniques, manual curation is not sufficient any longer [9]. A second focus of text mining research is to help in the interpretation of high-throughput screens such as gene expression or RNA interference screens, which typically generate large clusters of genes with somehow similar behavior. Identifying relationships within such clusters such as protein interactions or shared function is important to gain deeper insights. Text mining can also serve directly to cluster genes by phenotype [10]. In Lage et al. [11], for example, candidate genes for diseases are identified by clustering genes based on phenotype terminology extracted from a database with text mining. In addition to search and curation, knowledge extracted from the literature, combined with knowledge from databases, helps generating hypotheses, which can then be further verified. Examples for improving protein function prediction with results from text mining are given in Gabow et al. [12] and Groth et al. [10].

With ALI BABA, we provide means to efficiently search and browse PubMed citations, extract basic information, and link these to additional information available from relevant databases [13]. The basic idea behind ALI BABA is to display the contents of a collection of PubMed abstracts as a graph, that is, biomedical entities are nodes, and connections between those refer to potential associations, for example, interactions between proteins. ALI BABA, therefore, parses abstracts selected by the user for proteins, diseases, enzymes, and so on, and searches for potential relationships. The resulting graph should be understood as a summary of all abstracts, restricted to molecular biology entities and their associations. Figure 15.1 shows an example of such a graph, which resulted from 20 abstracts for the query "glutamate metabolism." Clicking on nodes and edges accesses the original text that contains them (see lower right panel in the figure). Each node is linked to one or more entries in a relevant biological databases; for instance, proteins are linked to UniProt and drugs to either DrugBank or MeSH.

In the remainder of this chapter, we will present the ALI BABA tool, starting with examples relevant to systems biology. We describe the functionality of ALI BABA from a user's perspective in Section 15.2. In Section 15.3, we give an overview of the techniques underlying ALI BABA and present quantitative assessments of the core techniques. We conclude the chapter with a discussion of related tools and future perspectives for biomedical text mining.

From the Web page `http://alibaba.informatik.hu-berlin.de/`, users launch ALI BABA via Java Web Start.[1] Installation instructions for this environment can be found on the Web page, although it is nowadays available on most systems by default. The Web page also provides a manual, further information, answers to frequently asked questions, as well as additional examples. As a convention for this chapter, we will write user queries to ALI BABA enclosed in double quotations marks ("query"), entities such as genes and diseases in italics (*Dickkopf*), and actions a user can take as well as items in ALI BABA in teletype (`File` menu).

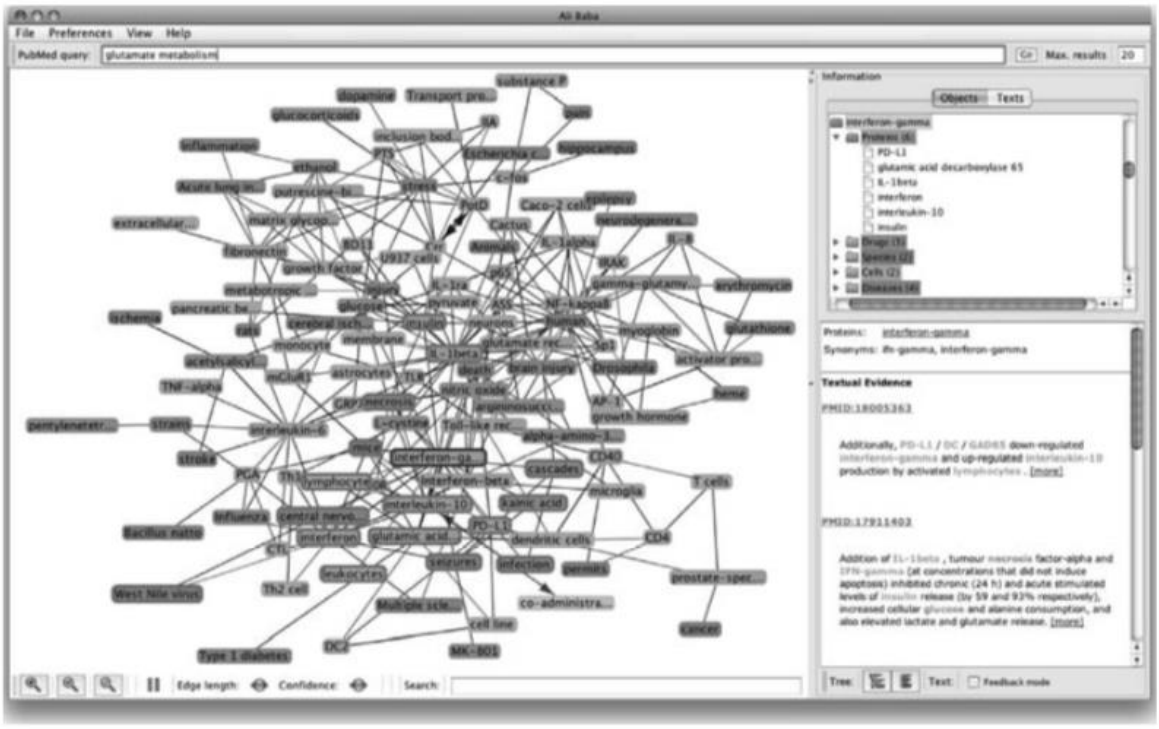

Figure 15.1 Ali Baba graph resulting from a query for "glutamate metabolism." Highlighted is the protein "interferon–gamma." Associated proteins and cells can be seen in the `Object` panel (upper right), with the supporting evidence being shown in the `Text` panel (lower right).

VALIDATION ISSUES IN REGULATORY MODULE DISCOVERY

Over the last decade, many new techniques have emerged for measuring the functional activity of genes. These include measurements of individual genes through their transcriptional activity, the interaction of their products, or by *in vitro* experiments with synthesized DNA. For each measurement technique a different set of data is created, each containing information on the functioning of genes, but under differing conditions and with different degrees of experimental error. At the same time, summary information of accepted gene behavior is being collected in the form of the gene ontology database and annotation terms for individual genes. These resources represent accumulated knowledge rather than individual experimental data. Current research is being undertaken to investigate ways in which these rich but diverse sources of information about gene behavior can be combined to provide a more accurate interpretation of experimental work. The goals are two-fold. Firstly, fusing data from diverse sources can be used to stabilize the results from individual experiments. For example, microarray experiments have been generally found to produce data of high variance, and therefore require some form of regularization before results can be interpreted. Secondly, when the data are processed to infer some

higher organization among genes, it is essential to have some form of validation of the results.

Work in molecular biology has focused both on identifying the function of individual genes and the way in which they interact in regulation processes. In nature, complex functions of living cells are carried out through the concerted activities of many genes and gene products that are organized into coregulated sets also known as regulatory modules [1]. Understanding the organization of these sets of genes will provide insights into the cellular response mechanism under various conditions. Recently, a considerable volume of data on gene activity, measured using several diverse techniques, has become widely available. By fusing these data using an integrative approach, it may be possible to unravel the regulation process at a more global level. Although an integrated model could never be as precise as one built from a small number of genes in controlled conditions, such global modeling can provide insights into higher processes in which many genes are working together to achieve a task. Various techniques from statistics, machine learning, and computer science have been employed by researchers for the analysis and combination of the different types of data in an attempt to understand the function of regulatory modules.

There are two underlying problems resulting from the nature of the available data. Firstly, each of the different data types (microarrays, DNA-binding, protein–protein interaction, and sequence data) provides a partial and noisy picture of the whole process. They need to be integrated in order to obtain an improved and reliable picture. Secondly, the amount of data that is available from each of these techniques is severely limited. To learn good models, we need considerable amounts of data. Unfortunately, data are only available for a few experiments of each type. These two problems are often cited as a reason for taking an integrative approach. However, integration will filter and obscure some of the information in the actual experimental results, and thus proper validation methods are required to test the effectiveness of any approach.

DATA TYPES

Various types of data are used to identify regulatory mechanisms. These are primarily generated by molecular biologists using experimental techniques. In most cases, a considerable amount of data processing must be applied before the results can be interpreted.

One of the most important sources of data is genome-wide measurement of mRNA expression levels carried out using microarrays. These have received considerable attention in the last 6 years and various technologies for microarray measurement have been developed [2]. Microarrays allow simultaneous measurement of the expression levels of a large number of genes. Similar expression profiles identify genes that may be controlled by a shared regulatory mechanism. An important point to note is that coregulation does not necessarily imply only positive correlation of expression values, as some of the genes might be downregulated, while others may be upregulated [3]. Processing microarray data to make different experiments as far as possible

comparable is known as normalization. A good overview of techniques for normalization and analysis is provided by Quackenbush [4] and a detailled discussion of the statistical issues involved is given by Smyth [5].

Spellman was one of the microarray pioneers who studied the global expression of genes [6]. He studied both the expression variation at various time points in the yeast cell cycle, and, along with other researchers [7], the response of the yeast genes when subjected to various kinds of stress.

A second major source of data is transcription factor–DNA binding data, which is generated as a result of the chromatin immunoprecipitation (ChIP) technique, also popularly known as the ChIP–chip assay. The technique is used to determine whether proteins, including transcription factors, will bind to particular regions of the chromatin within living cells. Harbison et al. determined the global genomic occupancy of 203 transcription factors in yeast, which are all known to bind to DNA in the yeast genome [8]. Lee et al. produced a similar yeast dataset for a smaller number of transcription factors [9]. Both these researchers reported results in the form of a confidence value (statistical P value) of a transcription factor attaching to the promoter region of a gene. The reason behind using statistical techniques was to reduce the experimental errors inherent in microarray technology and to account for multiple cell populations. One of the prominent problems with such approaches is that in order to infer whether a transcription factor is attached to the promoter sequence or not, we have to choose an arbitrary artificial threshold of the P-value.

Transcription factor binding motifs are sequence patterns observed in the intergenic regions of the genome usually located upstream of the genes. They are thought to be responsible for allowing access of transcription factors to binding sites. Initial approaches to identifying these were based on first clustering genes by coexpression and then looking for common sequences in the upstream regions of the genes located in the same cluster. Kellis et al. used comparative genome analysis between three related yeast species to find these motifs [10].

Protein–protein interaction (PPI) data for human and other organisms are available as a result of advances in technologies like mass spectroscopy and yeast two-hybrid assays. There has been a tremendous growth in this type of data in the recent years.

Conventional biological studies focus on one gene or one protein at a time. Life, however, is a complex system that is not subject such a reductionalist approach. In today's postgenomic era, biologists believe that many genes and proteins interact in various fashions and that the deciphering and modeling of interaction among them would help better reveal and understand the mechanisms of living systems [1]. The emerging field of systems biology attempts to investigate such complex biological interaction from a systems viewpoint instead of individual molecules or components. New computational techniques are much needed for this new scientific endeavor, and, in particular, imaging plays an important role of providing objective, repeatable, quantitative phenotyping measures for complementing and correlating with large-scale genotyping studies. In this chapter, we will discuss the computational imaging and modeling techniques used in systems biology studies.

Computational techniques in systems biology can be roughly categorized into two broad classes: bioinformatics and bioimage informatics. Bioinformatics in systems biology mainly focuses on the biomarker discovery, including high-throughput molecular data analysis, molecular networks reconstruction from high-throughput data, molecular networks analysis, and so on. Bioimage informatics, on the other hand, address issues of image phenotyping, secondary screening, target validation, drug lead selection, and so on. The two classes of techniques are not necessarily orthological and are often integrated in solving complex problems. For example, Figure 17.1 exemplifies a systems biologic oriented workflow for biomarker discovery and validation, involving both bioinformatics (left) and bioimage informatics (right). The biomarker can be identified directly from the high-throughput data, such as gene microarray and mass spectrometry, as well as from the integrated molecular networks, such as gene regulatory networks, protein–protein interaction networks, and metabolic signaling networks. The molecular networks can be reconstructed from the high-throughput biological data by computational modeling means and integrated with the existing knowledge from the literatures and the databases, such as (KEGG) Kyoto Encyclopedia of Genes and Genomes and (DIP) Database of Interacting Proteins. Once we have identified certain candidate biomarkers, the next step is to validate them by biological experiments, such as knockout experiments using RNAi (RNA interference) and PCR (polymerase chain reaction). The validation provides valuable feedback for the next iteration of biomarker discovery process. The biomedical imaging provides multidimensional functional and morphologic features of biological systems under investigation and plays an important role in the biomarker validation, for example, high content screening for *in vitro* experiments and molecular imaging for *in vivo* experiments.

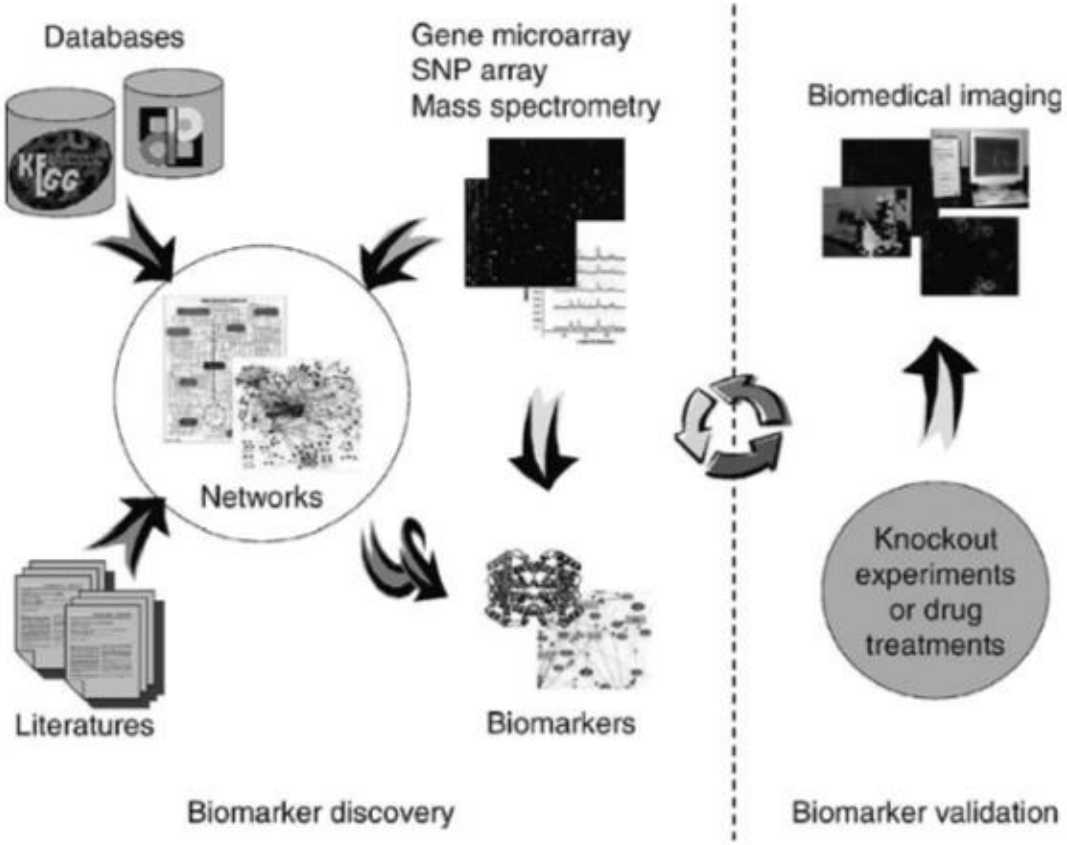

Figure 17.1 Systems biology approach that integrates bioinformatics and biomedical imaging in iterative biomarker discovery and validation.

CONNECTING BIOINFORMATICS AND BIOMEDICAL IMAGING

Information obtained from bioinformatics studies such as high-throughput genomics, proteomics, and metabonomics can be correlated with biomedical imaging to aid in the understanding of molecular interactions and disease pathways. For instance, cell-based screening assays can be used to distinguish between phenotypes and investigate interactions between signaling pathways and are useful in determining the interaction between drug candidates and target genes. Molecular imaging offers the possibility of imaging *in vivo* gene expression and protein–protein interactions. HCS can output screening hits and functional effectors. Starting from those effectors, we can study their interactions from a systems viewpoint such as that of metabolic networks. Metabolic networks can give biologists hints such as which genes/proteins/enzymes are in the pathway under study, they can then again use cellular imaging to validate them. In this section, we briefly discuss how to connect bioinformatics to biomedical imaging within systems biology framework and review some recent development of systems biology approach in this direction.