Chapter 8
Biological Network Modeling and Analysis

Sebastian Jan Janowski, Barbara Kaltschmidt, and Christian Kaltschmidt

Abstract Each scientist needs to be aware of the complexity of cellular life and the modeling possibilities to be able to reconstruct, analyze, and simulate biological systems. Bioinformatics modeling, analysis, and simulation are highly interdisciplinary disciplines using techniques and concepts from computer science, statistics, mathematics, chemistry, biology, biochemistry, genetics, and physics, among others. Without knowledge about these research topics, it is almost impossible to produce good theoretical models, which can be used for hypothesis testing. Therefore, this chapter gives an impression of what can be modeled from the bioinformatics and biological point of view and introduces into biological networks, common analysis techniques from graph theory, and possibilities to reconstruct, simulate, and share biological networks based on database content.

S.J. Janowski (✉)
Faculty of Technology, Department of Bioinformatics, Bielefeld University, Universitätsstraße 25, D-33501 Bielefeld, Germany
e-mail: janowski@cebitec.uni-bielefeld.de

B. Kaltschmidt
Faculty of Biology, Molecular Neurobiology, Bielefeld University, Universitätsstraße 25, D-33501 Bielefeld, Germany
e-mail: barbara.kaltschmidt@uni-bielefeld.de

C. Kaltschmidt
Faculty of Biology, Cell Biology, Bielefeld University, Universitätsstraße 25, D-33501 Bielefeld, Germany
e-mail: c.kaltschmidt@uni-bielefeld.de

M. Chen and R. Hofestädt (eds.), *Approaches in Integrative Bioinformatics: Towards the Virtual Cell*, DOI 10.1007/978-3-642-41281-3_8,

8.1 What Can or Should Be Modeled?

What is cellular life? The simplest answer from the biological point of view is the following: anything that contains DNA or RNA [1], shows self-organization, and has evolved over time as described by Manfred Eigen [2]. Motivated to seek a theory to understand life, many decades ago researchers embarked on the study of biological systems [3, 4]. Their main goal is not to imitate life but rather to understand the universal logic and properties of living systems. Cellular functions which do not rely on simple enumeration of molecular components and processes, such as transcription, translation, and modifications, are carried out constantly. These components never act as one independent element. Thus, present-day cellular biology is challenged to reconstruct coupled dynamical models with many differing elements and strongly interacting systems. Therefore, scientists endeavor to provide a new look at data on the present organisms to validate or reject hypotheses.

The main task for modern biology is to trace phenotypical properties back to specific molecules. Therefore, theoretical models are constructed, consisting of the formation of switching rules that obligate cell features. With modern systems biology and bioinformatics, those theoretical models are pictured. Therefore, natural sciences produce a holistic view of different levels of organizations. Using causal relations, theoretical models are constructed using several different switching rules. Through the turning on and off of one or more genes, as controlled by one or more molecules, the properties and dynamics of a cell can change. This can result in different cell behavior, where the concentration of some other molecule is altered, with the effect of turning on or off some other genes [1, 5].

Thus, to model and investigate cellular life, several different key components of real-life systems have to be considered. The central dogma of molecular biology stated by Francis Crick in 1958 describes the basic information flow in cells with the following sentence: “DNA makes RNA, which in turn makes Proteins” [6, 7]. In general, this statement is correct, whereas it is very simplified. Nowadays, natural science has investigated many processes and functions in detail, such as transcription, translation, and posttranslational modification, among others, which extend this stated dogma. The investigation of other regulatory processes, such as microRNA fine regulation, is still in their beginning phases. Table 8.1 gives an example of specific cell-type characteristics and dynamics to show the variety of living organisms [8].

Although all these presented aspects have to be considered in the modeling of a biological system and put into relationship with the biological dogma, it is neither recommended nor practical to model all aspects. Too many unknown parameters will come up, with the danger being that a fitted model will match to nearly anything. Fitted parameters can be even misleading or become meaningless. Furthermore, the larger the model, the longer it will take to determine parameters and to analyze properties of interest. Therefore, each model has to be limited to a practical size and linked to clear scientific questions.

Table 8.1 Biological cell characteristics for *E. coli*, yeast (*S. cerevisiae*), and mammalian (human fibroblast) based on [8]

Property	*E. coli*	Yeast (*S. cerevisiae*)	Mammalian (human fibroblast)
Cell volume	$\sim 1\,\mu m^3$	$\sim 1{,}000\,\mu m^3$	$\sim 10{,}000\,\mu m^3$
Proteins/cell	$\sim 4 \times 10^6$	$\sim 4 \times 10^9$	$\sim 4 \times 10^{10}$
Genes	~4,500	~6,600	~30,000
Size of regulator binding site	~10 bp	~10 bp	~10 bp
Size of promoter	~100 bp	~1,000 bp	$\sim 10^4$ to 10^5 bp
Size of gene	~1,000 bp	~1,000 bp	$\sim 10^4$ to 10^6 bp (with introns)
Diffusion time of protein across cell	~0.1 s $D = 10\,\mu m^2/s$	~10 s	~100 s
Diffusion time of small molecule across cell	~0.1 ms $D = 1{,}000\,\mu m^2/s$	~10 ms	~0.1 s
Time to transcribe a gene	~1 min (80 bp/s)	~1 min	~30 min (including mRNA processing)
Time to translate a protein	~2 min (40 aa/s)	~2 min	~30 min (including mRNA nuclear export)

One possibility to limit model size is by using biological networks. These networks can be restricted to only one -omic level, such as metabolomics or proteomics. The main advantage of biological networks is that they can be used to answer scientific questions with the focus on important regulatory elements, rather than building up whole systems.

8.2 Biological Networks

Cellular life is mostly a network of interacting elements. To visually represent and analyze the various interactions and relationships, biological systems can be modeled as biological networks, which are based on mathematical graphs (see Definition 1).

Definition 1. A graph is an ordered pair $G = (V, E)$:

- Comprising of a set V of vertices and a set E of edges, where each edge is assigned to two (not necessarily disjunct) vertices.
- The order of a graph is $|V|$, comprised of the number of vertices.
- The size of a graph is $|E|$, comprised of the number of edges.
- The degree of a vertex is the number of edges that connect to it and are defined by $N_G(v)$ or $N(v)$.

The objects, represented by nodes, are called "vertices" and the links, represented by directed or undirected arrows, are called "edges." In general, the smallest level of details is the molecular level, describing DNA, RNA, proteins, and metabolites interacting with each other. Thus, nodes can be any kind of biological compounds belonging to such a system. Edges are used to represent biological relations and processes, such as activation, inhibition, and expression, among others. To model all system elements, information flow, and dynamics, different biological networks were introduced as described in the following:

- **Transcription networks (or gene regulation networks)**
  Transcriptional networks control the gene expression within cells in time, space, and amplitude [9]. Usually these kinds of networks describe how one gene is controlled by the product of another gene. Therefore, the highly interconnected processes are modeled with a directed graph, in which nodes represent gene, transcription factors, and/or proteins and edges indicate mechanisms, such as transcription, DNA binding, protein synthesis, and degradation, among others. Furthermore, the synthesis of RNA, posttranscriptional events, mRNA turnover, and translation can also be considered. However, as these kinds of networks model a wide range of biological processes, they play a major role in protein-protein interaction networks, signal transduction networks, metabolic networks, and others, which are described in the following.
- **Protein interaction networks**
  In terms of the degree of regulation, it becomes apparent that a protein can never be investigated in isolation. Moreover, it has to be examined in the context of other proteins and their interacting network, in the so-called protein-protein interaction networks. The majority of biological processes within a cell are controlled and mediated by proteins [1, 5]. They interact with other molecules, such as low-molecular-weight compounds, lipids, and nucleic acids to ensure transcription, translation, splicing, mechanical strength, transport, immunity, signal transduction, growth, development, and many other processes. The types of interactions range from transient interactions, occurring for a limited time, such as they appear in protein kinases, protein phosphates, and others, up to static interactions, such as the transfer of biosynthetic intermediates between catalytic sites without the diffusion into the enzyme's surrounding. A further important aspect of protein-protein interaction is the signal transmissions from the external environment to specific locations within the cells.

  However, such protein-protein interaction networks enable the scientist to investigate protein functions, system dynamics, and biological mechanisms [9–15]. Reconstructing these kinds of networks, unknown proteins can be grouped into known biological context and important proteins into functional groups, subnetworks, and motifs identified and examined in detail. This kind of analysis has become so important and powerful that it already contributes to new therapeutic strategies [13, 16, 17].

- **Signal transduction networks**
  Signal transduction networks are of special interest in biological and medical sciences as many diseases are related to disturbances in signaling networks [18]. In general, signal transduction links intracellular processes to the extracellular environment of a cell. The general aim is to model and describe cellular functions in response to external stimuli. Therefore, information transmission is modeled, starting with the binding of extracellular ligands to receptors and resulting in cell response that triggers a cascade of signal transduction reactions. The sequence of reactions involved mainly relies on reversible chemical modifications and complex formations, such as phosphorylation. The final targets of the processes are transcription factors and metabolic enzymes. In summary, signal transduction pathways transform a set of inputs into a set of outputs.

  In contrast with other networks, such as protein-protein interaction networks, signaling networks are basically directed. From the topological point of view, the networks involve many different motifs, such as positive and negative feedback loops. One of the most prominent examples is the negative feedback loop of the transcription factor NF-κB [19, 20].
- **Metabolic networks**
  Metabolic networks have a fundamental importance in biochemistry and biotechnology, as many scientists modify or alter metabolic networks to produce fine chemicals, antibiotics, industrial enzymes, antibodies, etc. Furthermore, metabolic networks are used in biomedicine enabling a better understanding of metabolic mechanisms and for controlling infections. Therefore, scientists examine differences, synergies, and other interactions between human beings and pathogens. In general, the main goal of metabolic networks is the modeling of cellular processes, such as the uptaking and digesting of substrates from the environment, energy generation, growth, and cell survival, among others. Many of these networks are available online in databases, such as KEGG [21], EcoCyc [22], and BioCyc [23]. The networks refer to metabolites (amino acids, glucose, polysaccharides, glycans, etc.) and their biochemical reactions.
- **Correlation networks**
  Correlation networks represent statistical associations between variables derived from experiments, such as derived from whole genome arrays, mass spectrometry, and enzyme-based proteomic experiments, among others [9]. The global analysis approach is to give a broad overview of the state of the organism. Due to technological advances in systems biology, experimental approaches are able to provide qualitative and quantitative information, which can be used for comprehensive insights into biological systems.

  Usually the resulting datasets are mainly independent variable-unit entries. However, based on the experimentally measured values, correlations can be determined from either the probability point of view or the strength of variable units. The first approach measures if two values have a connection by coincidence or if there seems to be a real link. Therefore, correlation coefficients are

calculated expressing the connection probability. The accuracy of this approach mainly depends on the sample size of the experiment. Examining a large number of samples increases the probabilities for finding real connections and, moreover, increases the probability of identifying whether weak connections are true. The second approach only considers connection from the strength of variable units, instead of the sampling size. However, an experimental validation based upon the results is the best way to confirm a predicted correlation.

- **Neuronal networks**
  In neuronal networks artificial neurons are connected to each other. The aim is to reconstruct systems as they appear in real life [24,25]. Thus, connections between neurons are modeled with neuronal summation, in which potentials and electric gap junctions define firing strategies and signal transduction from one neuron to another. In neuronal networks, neurons only respond to a subset of mostly simple stimuli given by their neighbors, whereas, in real systems, the information flow is based on inhibitory postsynaptic potentials and excitatory postsynaptic potentials. The modeling and analysis of neuronal networks has attracted wide interest in life sciences. For example, the subject of one application field is to model systems which are able to learn complex patterns and therefore build a kind of artificial intelligence.
- **Phylogenetic networks**
  Phylogenetic networks describe the evolution and relationship between different organisms. Usually, phylogenetic reconstructions are presented by trees rather than networks, in which branch points represent the evolutionary separation of two organisms. However, trees do not consider vertical and horizontal gene-transfer events. Thus, phylogenetic networks describe evolutionary processes in more detail. Kunin et al. give one prominent example of such a phylogenetic network in their article "The net of life: Reconstructing the microbial phylogenetic network" [26].
- **Ecological networks**
  Ecological networks typically present food webs. Food webs are limited representations of real ecosystems describing ecological communities focusing on trophic interactions between consumers and resources ("what eats what") [27–29]. In general, two trophic categories exist, called trophic levels. The first ones are the autotrophs, which produce organic matter from inorganic substances. The second level, the heterotrophs, obtains organic matter by feeding on autotrophs and other heterotrophs. It is a unified system of exchange, adopted to analyze interrelationships between community structure, stability, and ecosystem processes.

  The analysis of food webs has shown that the evolution of realistic food web structures can be explained on the basis of simple rules regarding population abundance and species occurrence. For example, ecologists and mathematics have figured out early on that the structure of food webs consists of nonrandom properties, such as scaling laws. By examining a predator-prey model (resource-consumer, plant-herbivore, parasite-host), it becomes obvious that the size of one species is crucial to the stability of the whole system [30].

However, food webs are an important representation for the prediction of ecological events. They are mainly used to understand biological systems and moreover to protect them from outside influences, such as climate change, foreign wild species, and the narrowing of the habitat.

Summarized, the presented biological networks are able to capture all -omic levels and, furthermore, able to model ecological events and other correlations. With these advantages bioinformatics and systems biology have a set of powerful integrated frameworks to present, integrate, and visualize knowledge. Furthermore, graph theory comes with powerful approaches to analyze those networks as described in the following.

8.3 Biological Network Analysis Based on Graph Theory

As mentioned in the previous section, graphs or networks can be used to model many types of biological relations, biological processes, and biological questions. Furthermore, geometry and topology can give important clues about organization and information flow within a system. Graph analysis can determine structural properties of a network. Furthermore, graph theory can analyze vertex degrees, path lengths, diameter, and many other structural properties.

In general, graphs can have different types as presented in Fig. 8.1. In a **directed graph** an edge between the vertices u and v is represented by the ordered pair (u, v) [31]. Visually the ordered pair represents the direction of the arrowhead. However, there is a big difference between directed and undirected graphs for a given number

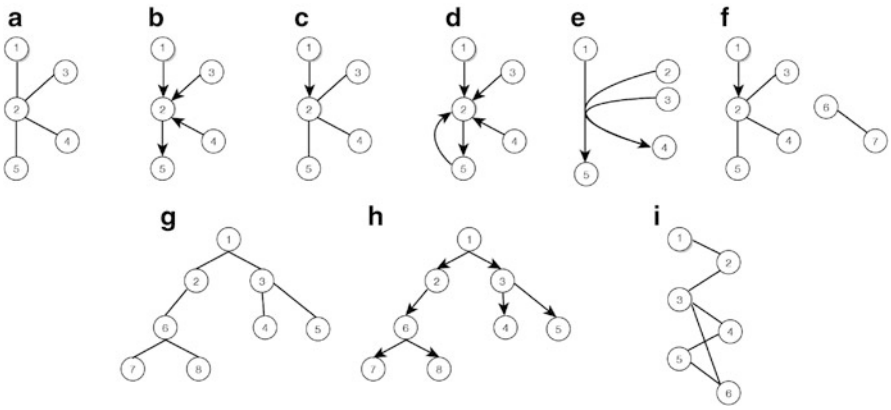

Fig. 8.1 Different graph types as they may appear in biological networks: (**a**) undirected, (**b**) directed, (**c**) mixed, (**d**) multigraph, (**e**) hyper-graph, (**f**) unconnected graphs, (**g**) tree, (**h**) rooted tree, and (**i**) bipartite graph

of vertices. The **amount of directed graphs** $N_{\text{dir}}(V)$ with V vertices is much higher than the amount of possible undirected graphs $N_{\text{undir}}(V)$ [9]:

$$\frac{N_{\text{dir}}(V)}{N_{\text{undir}}(V)} = 2^{\frac{(V^2-1)}{2}} \tag{8.1}$$

A **mixed graph** has both directed and undirected pairs. In the biological context it can represent protein-protein interaction networks, where some interactions are undirected, such as protein-complex bindings, and some interactions, such as activation, phosphorylation, and other processes are directed. A **multigraph** contains multiple edges, where two or more edges are incident to the same two vertices. A **hyper-graph** is characterized by more than two elements, which are connected to one interaction. Hyper-graphs are often used to model metabolic networks where several substances are used in one reaction to produce another substance.

A graph is **bipartite** if there is a partition of its vertex set $V = S \cup T$, such that each edge in E has exactly one end vertex in S and one end vertex in T. A **tree** is an undirected, acyclic graph, where vertices with only one edge are called leaves. All other vertices are inner vertices. The depth of such a tree is the length of the path from the root to a vertex. The height is the maximal depth. A rooted tree is often regarded as a directed graph [31].

A **subgraph** $G' = (V', E')$ of the graph $G = (V, E)$ is a graph where $V' \in V$ and $E' \in E$ [31]. The **density** of a graph is given by

$$\frac{2 \mid E \mid}{\mid V \mid (\mid V \mid -1)} \tag{8.2}$$

This definition indicates how dense or connected a graph is determining vertex degrees [32].

Two graphs G and G' are **isomorphic** $G \simeq G'$, if there exist a bijection $\varphi : V -> V'$ between the vertex sets of G and G', such that any two vertices u and v of G are adjacent in G if and only if (u) and (v) are adjacent in G', based on $xy \in E \Leftrightarrow \varphi(x)\varphi(y) \in E' \; \forall x, y \in V$ [31].

Global network properties are topological entities, such as distance, average path length, and diameter. A **path** is a sequence $(v_0, e_1, v_1, e_2, \ldots, v_{k-1}, e_k, v_k)$ of vertices and edges. The **length of a path** is given by its number of edges. The **distance** between two vertices is given by $d_G(u, v)$. A **shortest path** between two vertices is a path with minimal length d_{ij}. The **average path** length is defined by $d = \langle d_{ij} \rangle$. The **diameter** is defined by $d_m = \max(d_{ij})$, which represents the maximum path length. The correlation between edges and vertices is given by $\varepsilon(G) := |E|/|V|$ [31,32].

An **Eulerian path** is a path which contains every edge exactly once. A graph is an **Eulerian graph** if it contains an Eulerian path [31]. A path in an undirected graph that visits each vertex exactly once is called a **Hamiltonian path**. A graph that contains a Hamiltonian path is a **Hamilton graph** [31].

Going further into detail, vertex degrees and other topological indices are described in the following, which serve as a base for centrality measurements. Network centralities are a common method to determine important elements within a system. In the social sciences it is a common task to model relationships with graphs and, based on that, to identify people that are more influential than others. Similar questions can also be asked of biological networks.

A **centrality** is defined by the function $\mathscr{C} : V \mapsto \mathbb{R}$ on a directed or undirected graph $G = (V, E)$, which assigns a real number to every vertex (vertex degree). If one vertex is more central than another one, then $\mathscr{C}(v_1) > \mathscr{C}(v_2)$ is given [33].

A **vertex degree** $\delta_G(v) = \delta(v)$ is the number of edges $|E(v)|$ incident to the vertex, with loops counted twice. The **minimum degree** is characterized by $\delta(G) := \min\{d(v) \mid v \in V\}$, the **maximum degree** by $\Delta(G) := \max\{d(v) \mid v \in V\}$, and the **average degree** by:

$$d(G) := \sum_{v \in V} \frac{d(v)}{|V|} \tag{8.3}$$

The relation between the degrees is given by $\delta(G) \leq d(G) \leq \Delta(G)$ [9, 31, 32].

However, centrality measurements are only comparable inside the same network, and some measurements can only be applied on connected networks. One of the first centrality measurements is the **degree centrality**, defined by

$$\mathscr{C}_{\text{deg}}(v) := |e|e \in E \wedge v \in e| \tag{8.4}$$

This measurement counts the number of edges connected to a vertex. In several studies, this measurement was used to identify essential elements within a biological network. A study on *Saccharomyces cerevisiae* revealed that proteins with a high degree centrality are more essential in comparison to others [34]. Other studies described similar findings with degree centralities as described by Hahn et al. [35].

The **average neighbor degree** is defined by Junker and Schreiber [9]

$$k_{i,nn} = \frac{1}{k_i} \sum_{j=1}^{N_v} A_{ij} k_j \tag{8.5}$$

for each vertex n_i over all vertices N. A is the adjacency matrix of the graph G.

Further centrality measurements are stated on network paths. They give information about the importance of certain paths by using information about path length. The first presented measurement is called eccentricity centrality. For every vertex it determines the maximum distance to all other vertices. The vertex with the shortest paths to all other vertices is the vertex with the highest eccentricity value. Formally, the **eccentricity centrality** is defined as [36]

$$\mathscr{C}_{\text{ecc}}(v_1) := \frac{1}{\max\{\text{dist}(v_1, v_2) : v_2 \in V\}} \tag{8.6}$$

The second important centrality measurement is the **closeness centrality**, which assigns a vertex v a high value if the shortest path distances for all other vertices to v is minimized. Formally, it is defined as [37]

$$\mathscr{C}_{\text{clo}}(v_1) := \frac{1}{\sum_{v_2 \in V} \text{dist}(v_1, v_2)} \tag{8.7}$$

The **shortest path betweenness centrality** measures the ability to monitor communication between other vertices. These vertices, which are on the shortest paths between all other vertices, are the most relevant ones. Let $\sigma_{v_1 v_2}$ be the number of shortest paths between v_1 and v_2, whereas more than one shortest path can exist. $\sigma_{v_1 v_2}(w)$ denotes the number of shortest paths, including w as an interior vertex which is neither start nor end vertex of the paths. The communication rate is given by

$$\delta_{v_1 v_2}(w) := \frac{\sigma_{v_1 v_2}(w)}{\sigma_{v_1 v_2}} \tag{8.8}$$

If no shortest path between v_1 and v_2 exists, then $\delta_{v_1 v_2}(w) := 0$. With these definitions the shortest path betweenness centrality can be defined as [38]

$$\mathscr{C}_{\text{spb}}(w) := \sum_{v_1 \in V \wedge v_1 \neq w} \sum_{v_2 \in V \wedge v_2 \neq w} \delta_{v_1 v_2}(w) \tag{8.9}$$

A further centrality measurement is based on the eigenvector. It is used on strongly connected graphs such as protein-protein interaction networks, to determine essential elements within a network. The **eigenvector centrality** is the eigenvector C_{eiv} of the largest eigenvalue $\lambda_{\max}$ in absolute value of the equation system $\lambda C_{\text{eiv}} = A C_{\text{eiv}}$, where A is the adjacency matrix of the graph G [39].

The **clustering coefficient**, a basic measurement for the local cohesiveness of a network, measures the probability that two vertices with a common neighbor are connected. In the case of undirected graphs, there exist $E_{\max} = k_i(k_i - 1)/2$ possible edges between neighbors. The clustering coefficient C_i of the vertex n_i is then given as the number of edges E_i between the neighbors to the maximal number $E_{\max}$ with [9]:

$$C_i = \frac{2E_i}{k_i(k_i - 1)} \tag{8.10}$$

The **matching index** quantifies the similarity between two vertices on the number of common neighbors. The index is based on the following definition [9]:

$$M_{ij} = \frac{\sum \text{common neighbors}}{\sum \text{total number of neighbors}} = \frac{\sum_{k,l}^{N} A_{ik} A_{jl}}{k_i + k_j - \sum_{k,l}^{N} A_{ik} A_{jl}} \tag{8.11}$$

Table 8.2 For a given network size, many different graphs can be reconstructed, where the difference between isomorphic and non-isomorphic graphs is significant

Nodes	Number of connected isomorphic graphs	Number of connected non-isomorphic graphs
3	8	2
4	64	6
5	1,024	21
6	32,768	112
7	2,097,152	853
8	268,435,456	11,117
9	68,719,476,736	261,080
10	35,184,372,088,832	11,716,571

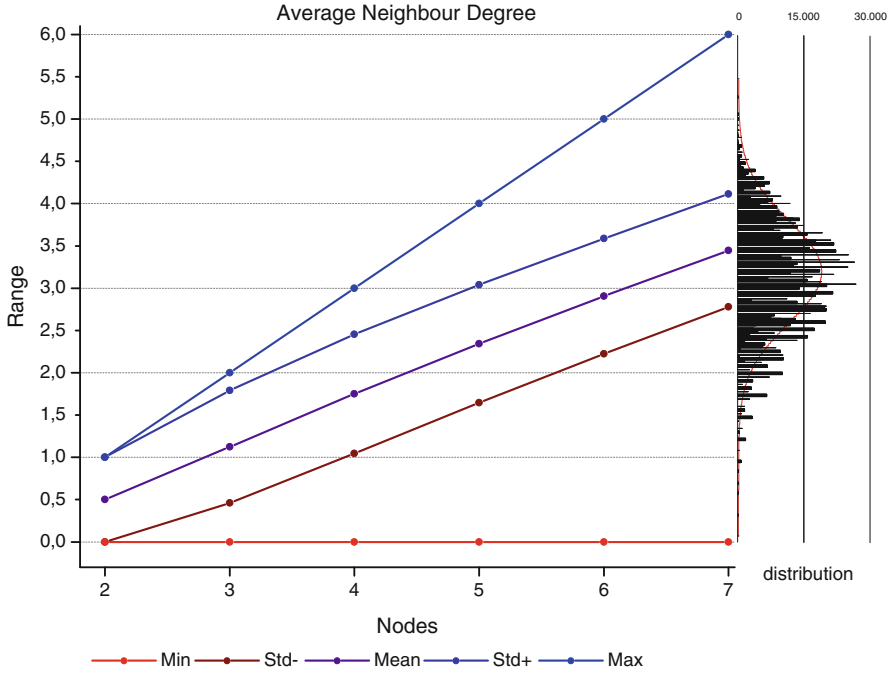

Fig. 8.2 The analysis of the distribution of graphs with the same average neighbor degree resembles a Gaussian curve, where thousands of different networks share the same average neighbor degree. The conclusion is that one specific average neighbor degree cannot characterize a unique network type [40]

In summary, all presented measurements are able to identify important elements within a graph. However, without a clear scientific question, the presented approaches can be misleading. Furthermore, scientists need to have in mind that a large set of graphs can share the same graph topological values [40]. In general, the number of possible graphs for a given node size is very large as presented in Table 8.2 [41]. Based on the non-isomorphic graphs, it was examined how many graphs share the same graph topology. Figure 8.2 presents the distribution of graphs with the same topological values.

Inferentially, thousands of different graphs share the same topological values. And having in mind that the discussed and examined graphs in biology have, in most cases, more than 30 nodes, the number of different graphs with the same topological values increases dramatically. Thus, graph theory has to be very carefully considered and only applied when it is linked to a specific scientific question. However, based on the presented definitions, a variety of analysis techniques are possible. The approaches enable structural as well as individual node analysis. Thus, it is not surprising, that applied to biological networks, it has become an important aspect in systems biology, bioinformatics, and theoretical biology [9].

8.4 How Biological Networks Can Be Modeled and Simulated

Modeling biological phenomena with the use of computer applications has become a common task. Therefore, different modeling techniques exist to study and analyze the dynamic details of biological systems. In general, biologists are more familiar with mathematical modeling, whereas computer scientists are accustomed to computational formalism. However, several approaches provide mathematical as well as computational capacities. In order to give an overview of existing modeling languages, the most important techniques in systems biology and biological network modeling are briefly described in the following subsections.

8.4.1 Ordinary Differential Equations

One of the most powerful techniques in modeling system dynamics is ordinary differential equations (ODEs), which provide a theoretical framework for discrete, continuous, deterministic, and stochastic models. In general, they describe the change rate of variables in the modeled system as a function of time. ODEs have been applied and used in many application cases and proved themselves very useful [8, 42, 43]. Furthermore, ODEs can be used to model entire systems with given kinetics [44, 45]. One common example for modeling gene activation or positive control is the Hill function in which the equilibrium binding of the transcription factor to its site on the promoter is modeled from zero to its maximal saturated level with Definition 2 (see Fig. 8.3 for a graphical representation).

Definition 2. A Hill function is defined by $F(X^*) = \dfrac{\beta X^{*n}}{K^n + X^{*n}}$, where:

- K is termed as the activation coefficient.
- β the maximal expression level of the promoter.
- n the steepness of the input function (the larger the n is, the more steplike the curve).

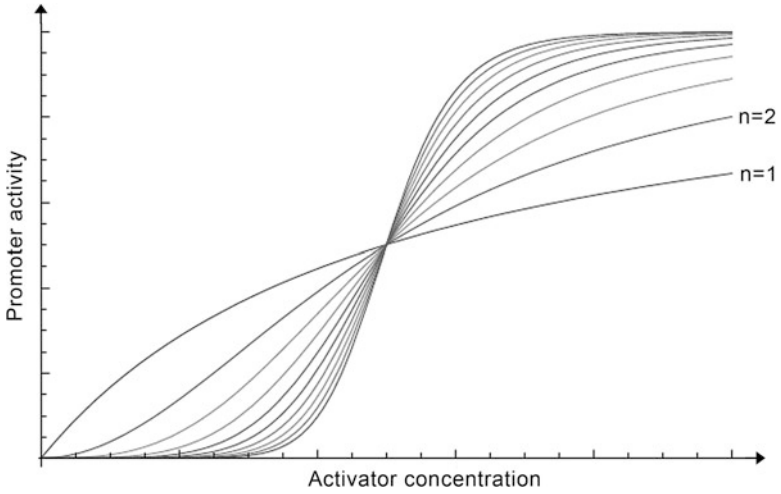

Fig. 8.3 Graphical plot of one Hill function with different steepness parameters (n) for the modeling of gene activation and positive control in biology

However, the model reconstruction with ODEs has some major drawbacks when the kinetic system parameters involved are unknown. With increasing network size and complexity, it becomes almost impossible to estimate all missing parameters. Due to high-throughput techniques, a huge amount of qualitative data is available, but the parameter estimation still remains challenging. Furthermore, precise quantitative measurements for parameter estimations are difficult to parametrically explore. A further disadvantage of ODE network modeling and analysis is that ODE-based models do not support any detailed insights into signal and information flow within biological networks. Thus, information flow, biological cascades, and system dependencies cannot be examined in detail.

8.4.2 Object-Oriented Modeling

Object-oriented modeling is a paradigm in which a system is primarily modeled with a set of related, interacting objects and the functions and services they provide [46]. These objects represent all entities relevant to the application (see Fig. 8.4 for an example). Nearly anything can be an object, which is defined as an assembly of classes. A class is a discrete reusable code block that has attributes, takes variables, performs functions, and returns values, among others. In general, objects do not exist in isolation from another. The relationships between the objects represent a wide set of different connections and interactions, for example, how one protein is related to a gene, or how one protein changes the state of another protein by phosphorylation. However, the modeling task is always specified for one specific

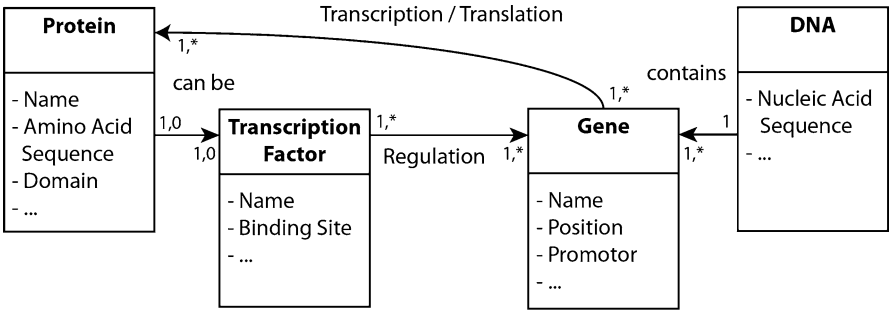

Fig. 8.4 An example of an object-oriented model in molecular biology. The model is focused on a mandatory set of properties, whereas a complete model is made up of more attributes and relationships. However, here, a protein can be a transcription factor regulating one or more specific genes. One gene can be even regulated by more than one transcription factor. The genes are derived from the class DNA, which contains a set of genes. Each gene alone or in combination with others can be transcribed and translated into one or more proteins. Each class is characterized by specific attributes, such as binding sites and nucleic acid sites, which are necessary for biological functions and molecular processing

context, where objects belong to each other and share a set of properties and methods to imitate the real-world system [47–49]. Using the standardized Unified Modeling Language (UML) [50], the object-oriented models can be made visually accessible through a set of graphic notation techniques.

8.4.3 Rule-Based Models

Rule-based specifications and formal grammars play an important role in the creation of photorealistic virtual organisms. Particularly plants and scientific models of vegetation structure are modeled with rule-based models [51]. One widely used formalism is the Lindenmayer system, a parallel rewriting system on strings. Based on an alphabet of symbols, a finite set of rules for string manipulations, a start string called axiom, and a mechanism to visualize data, it is possible to model the morphology of a variety of organisms. With an iterative process, which expands the model with new structures in each time step, growth processes can be modeled and simulated.

For example, having the axiom A and the rules $A \rightarrow B$ (letter A will be transformed into letter B) and the rule $B \rightarrow AB$ (letter B will be transformed into substring AB), a new string is generated in each time step by applying the aforementioned rules. Based on the system settings the development sequence for this model is described by $A \rightarrow B \rightarrow AB \rightarrow BAB \rightarrow ABBAB \rightarrow BABABBAB \rightarrow \ldots$. Finally, the expanded string only needs to be visualized to see developmental growth. In order to visualize this model, additional geometric rules have to be defined, which reconstruct geometric structures based on the appearance

and order of the letters in the development sequence. One of the first examples of branching structures generated by an L-system was given by Prusinkiewicz and Lindenmayer in 1990 [52].

8.4.4 Constraint-Based Models

Constraint-based models are mainly used for cellular metabolism. The main idea of this approach is to describe detailed dynamic models with a set of constraints which characterize the models' possible behaviors. Therefore, stoichiometric, thermodynamic, and enzyme capacity constraints are defined. Instead of single solutions, a set of possible solutions represents different phenotypes which comply with the constraints. Thus, models can comprise thousands of reactions, such as the metabolic reconstruction of the bacterium *Escherichia coli*, where 2,583 constraint reactions were defined [53]. Furthermore, these models and constraints can be used for other metabolic engineering applications. However, the classical constraint-based models focus at flux balance analysis of metabolic networks [54, 55].

8.4.5 Interacting State Machines

Interacting state machines are mathematical models for the description of temporal behavior within a system. The model is based on the states of its parts and not on its components. Therefore, hierarchies are expressed by diagram-based formalisms. Each of the parts can be in one of a finite number of states, whereas the machine is in only one state at a given time. However, by initiating a trigger event, the machine can change its condition. The main advantage of interacting state machines is that they require little quantitative data, as they model biological behavior in a qualitative way [56, 57]. Usually, models described with interacting state machines are used for model checking and interactive execution.

8.4.6 Process Algebras

Process algebras are used for the modeling of concurrent systems. The language provides a framework for the high-level description of interactions, communications, and synchronizations using a set of process primitives. Operators are used to combine these primitives. Therefore, this approach provides algebraic laws for the manipulation and analysis of process expressions using equational reasoning. In most of the cases, process algebras are used in signal processing, as presented in the work of Danos and Laneve. The authors introduced a protein algebra to demonstrate how standard biological events can be expressed in simplified signaling pathways [58].

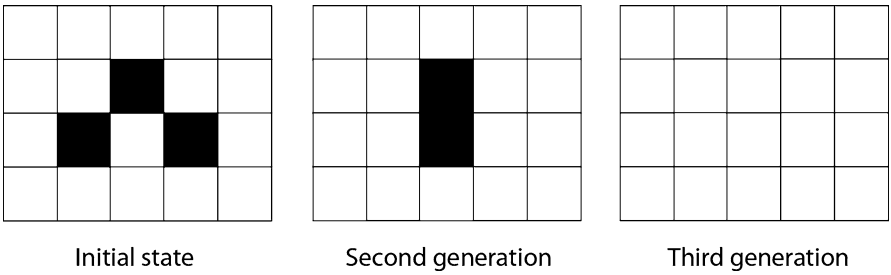

Fig. 8.5 An example of a simple cellular automaton with rules and settings of the "Game of Life" approach by John Horton Conway. From *left* to *right*: initial state and configuration (generation 1), second generation, and third generation

8.4.7 Cellular Automata

Cellular automata (CA) are used to model and simulate biological self-organization. They use a paradigm of fine-grained, uniform, parallel computation, which was used in many aspects of developmental biology [59–61]. With CA whole population dynamics can be simulated in which each individual's fate is dependent on its neighbor's behavior and existence. Therefore, a set of simple rules is defined that mimics the physical laws of the given system. The evolution of a CA is determined by its initial state, requiring no further input. The simulation is discrete in time, space, and state and, once running, evolves with its own given rules.

The most prominent example of a CA is the "Game of Life" devised by the British mathematician John Horton Conway in 1970 [62]. The example is based on a simple deterministic CA consisting of a regular two-dimensional grid of cells, in which each cell has a certain state: alive or dead. Every cell interacts with its neighbors based on the set of applied rules at each time step (see Fig. 8.5).

The following rules are applied to the "Game of Life" to calculate and simulate next generations:

- Any living cell with less than two living neighbors dies because of under population.
- Any living cell with two or three living neighbors does not change in the next generation.
- Any living cell with more than three living neighbors dies due to overcrowding.
- Any dead cell becomes alive by reproduction, when exactly three neighbors are alive.

Those rules are applied repeatedly to create further generation. Finally after n generations, a picture results that describes population structure, dynamics, population features, and system robustness, among others.

a

	Mutation probability	Mutation probability	Mutation probability
Grandfather		1	1
Father			1
Son	0.001	0.025	0.5

b

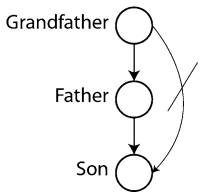

Fig. 8.6 A Bayesian network example from classical genetics studying mutations. (**a**) The probability that the son has a mutation is 0.001. If we know that his grandfather has the same mutation, the probability increases to 0.025. Thus, their genotypes are clearly dependent. But if we also know that his father has the mutation as well, the son's probability increases to 0.5. This additional information indicates that his father, independent of whether his grandfather has or does not have the mutation, only affects the son's probability. Therefore, only one conditionally network can be reconstructed (**b**), which matches the experimental data. All other possible networks are disregarded

8.4.8 Agent-Based Systems

Agent-based systems are similar to the concept of cellular automata, focusing on complex system behavior, structures, and phenomena in dynamics. This approach describes and simulates operations and interactions of autonomous agents in a given space. System operations and interactions are based on simple rules. However, in contrast to CAs, the agents are not placed on a grid or any similar environment. Moreover, the autonomous agents can freely move within the given 2D or 3D space. The most prominent examples are from multicellular studies, such as tumor growth studies [63], morphogenesis [64], and immune response [65].

8.4.9 Bayesian Networks

A technique for biological network modeling is the so-called "Bayesian networks" theory. Bayesian networks are used for the automatic reconstruction of causal signaling network models from experimentally derived data [66–68]. The core of this approach is the notion of conditional independency. This approach calculates probabilistic relationships to estimate which network structures, circuits, and motifs can be derived from the given biological data. This results in one or a set of possible directed acyclic graphs that match the experimental data conditions best. Nodes, which are not connected within the graph, represent variables which are conditionally independent. Nodes that are connected to each other represent strong probabilistic relationships based on experimental conditions. One example of such an approach is presented in Fig. 8.6.

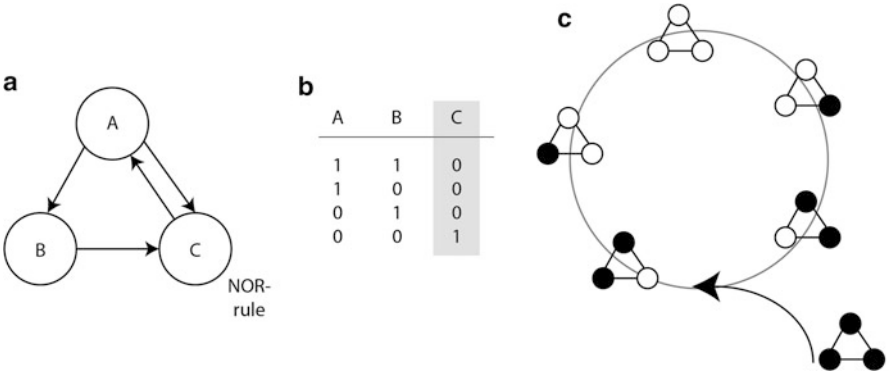

Fig. 8.7 A possible Boolean network based on three nodes (**a**), each having a state 0 (OFF) or 1 (ON). The states for each node are determined by the input of the other nodes. Nodes 1 and 2 copy their single input, while node 3 performs the Boolean function NOR on its inputs as described in the table (**b**). The dynamic system is described in (**c**), where filled nodes are on and lights are off

However, the reconstruction of such networks demands a large number of datasets. The greater the network, the larger the necessary experimental datasets must be. Otherwise, probabilistic relationships and independencies cannot be determined.

8.4.10 Boolean Networks

In 1969, Boolean networks were introduced by Kauffman to model gene regulatory networks [69]. Here, genes are modeled by Boolean variables which represent their active and inactive states within the model. A Boolean network is a directed graph, where all nodes are equivalent and receive information inputs from their neighbors. Every node can only take two binary values, 0 (OFF) and 1 (ON). These values represent the dynamic activity and behavior of the involved elements. Information flow and statement acting is determined by a logic rule. Therefore, the logical operators and, or, and not are used. If the statement is true, the logical operation results in an ON state; otherwise it remains in the OFF state (an example is given in Fig. 8.7).

The main advantage of this technique is the reduced number of parameters necessary while still capturing network dynamics and producing biologically predictions and insights [70]. However, quantitative measurements cannot be included for precise predictions and analysis.

8.4.11 Boolean Formalization

This approach formalizes in Boolean terms genetic situations for the description of complex circuits [71–73]. The main goal of this language is to formalize a

complex model in a compact and unambiguous way by functions of binary variables. Therefore, three different types are defined and used. The genetic variable describes the gene state, being normal or mutated, and the recognition site, being a promoter, operator, terminator, or other. The environment describes temperature and the presence of different substances. Internal variables are used to memorize previous system states at a given time. Associated functions calculate the proceeding periods of the system with regard to the present variables. In order to reduce the algebraic expressions to its simplest form, tabulations of the logic equations as Veitch matrices are used. The Veitch matrices give a clear and exhaustive view of all calculated system states and show which states are stable and how the model proceeds from state to state.

8.4.12 Petri Net

A Petri net is a mathematical modeling language for the description and analysis of complex and distributed systems. Therefore, it provides an exact mathematical definition of its execution semantics. The language was introduced by Carl Adam Petri in 1962 [74] and constantly developed. Thus, this language comes with a well-developed mathematical theory for process analysis.

Reisig et al. presented the first basic definition in their article "A Primer in Petri Net Design" in 1982 [75]. This resulted in the general formalism presented in Definition 3.

Definition 3. A basic Petri net is defined by the tuple $\mathrm{PN} = (P, T, F, W, m_0)$, where:

- $P = \{p_1, p_2, \ldots, p_n\}$ is a finite set of places.
- $T = \{t_1, t_2, \ldots, t_n\}$ is a finite set of transitions.
- P and T are pairwise disjoint.
- $F \subseteq (P \times T) \cup (T \times P)$ is a set of arcs from places to transitions and transitions to places, where $(p_i \rightarrow t_j)$ denotes the arc from place p_i to transition t_j and $(t_j \rightarrow p_i)$ the arc from transition t_j to place p_i,
- W is the weight function $(W : F \rightarrow \mathbb{R})$ which assigns every arc a non-negative integer, where $(f : p_i \rightarrow t_j)$ denotes the weight of the arc from place p_i to transition t_j.
- m_0 is the initial marking $\forall\, p_i \in P$.

A Petri net is based on a directed bipartite graph, in which the nodes represent transitions and places. Regarding the graphical representation, places are drawn as circles, transitions are drawn as rectangles, and arcs are drawn as directed arrows. The directed arcs describe which places are pre- and/or post-conditions for which transitions. Each place can contain tokens, which are drawn as black dots. The start configuration of a Petri net model is described by the state m_0, which assigns tokens to each place. With this graphical notation, processes such as choice, iteration, and concurrent execution can be modeled stepwise and analyzed (see Fig. 8.8).

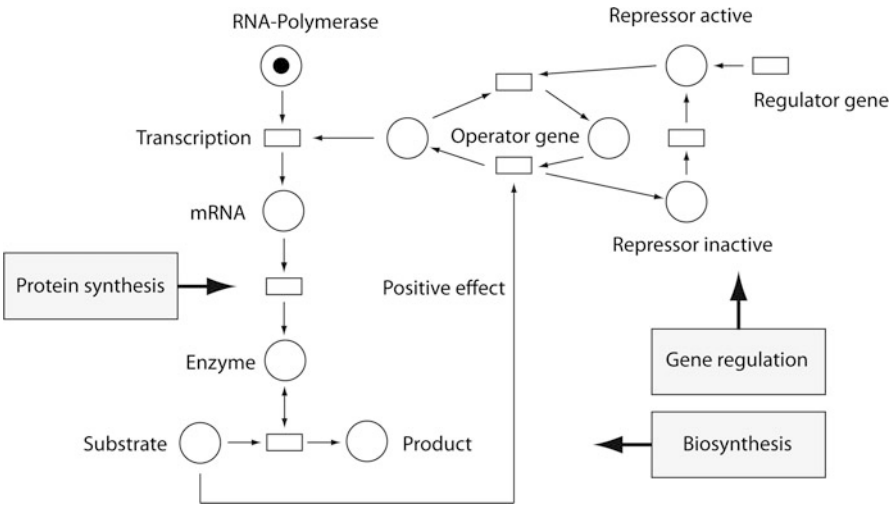

Fig. 8.8 The possibility of modeling abstract biological processes with Petri nets. The model is based on gene-controlled biochemical reactions, such as gene regulation and protein synthesis

Due to the presented formalism, Petri nets stand out by their balance between modeling power and analyzability in comparison to other modeling techniques. Furthermore, concurrent systems can be automatically determined, although some of the systems are difficult and expensive to determine [76]. Thus, the various modeling possibilities and analytic power of the proposed formalism offer a well-developed basis for the description of chemical processes and a mathematical theory for process analysis.

8.4.13 Visual Modeling

A further way to model a biological system is by using a standard graphical notation, such as the Systems Biology Graphical Notation (SBGN) [77]. SBGN is a visual language which focuses on the graphical notation of biological networks. It provides a common notation to represent interactions and regulations between molecular species, such as binding, complexation, and protein modification, among others. It consists of three complementary languages: process diagram, entity relationship diagram, and activity flow diagram. Together the different notations enable scientists to represent biological networks in a standard and unambiguous way (see Fig. 8.9 for an example).

In summary, each modeling technique comes with specific features and constraints. In order to model and analyze a biological system a powerful theoretical framework is necessary. Thus, visual languages such as SBGN are not suitable for

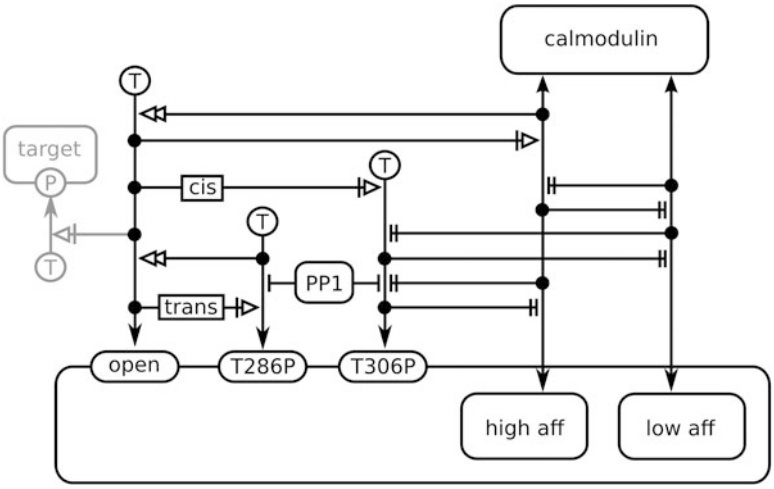

Fig. 8.9 SBGN entity relationship diagram representing the effect of calmodulin binding on CaMKII activity, using the nested entities of ER L2 V1 [78]

systems biology analysis, as they do not provide any kind of analytical environment. Furthermore, these languages consider only a limited graphical representation of the biological components. Object-oriented models are software-intensive and complex systems. As systems evolve, classes and the function they perform need to be changed more often. This can result in a schema, where complexity continuously grows. Thus, a clean programming, organization, and notation are necessary during model design and software implementation. Furthermore, well-defined interfaces between objects are mandatory to keep the model maintainable. Otherwise, model parameters can become distorted or even incorrect. Ambiguities in data flow can also occur. Therefore, the following review only focuses on modeling techniques that provide sophisticated analysis power and are clean and well defined in their semantics. To show how often and in which application cases the aforementioned techniques are used, Machado et al. summarized literature references, classified by the type of biological process [79] (see Table 8.3). Boolean formalizations are not considered in this review as this approach is frequently used in systems biology and bioinformatics. Furthermore, the same or similar results can be produced with Boolean networks, ODEs, or Petri nets, among others.

The first thing to point out is that all formalisms have been applied to signaling networks. This is not surprising, as signaling networks have the largest number of features, such as spatial localization, multistate components, network information flow, and robustness, among others. Therefore, each of the presented formalisms contributes with powerful features. A smaller number of formalisms are applied to metabolic networks. However, this does not indicate that other formalisms are not able to model those systems. Moreover, it seems that Petri nets, process algebras,

Table 8.3 Overview of the amount of literature references using the presented formalism classified by the type of biological process [79]. Based on the evaluated information, signaling networks have been modeled and analyzed with all formalisms. Gene regulatory networks and metabolic networks have only been modeled with specific techniques due to their specific system dynamics and topology. However, differential equations, constraint-based models, and Petri nets have been used as universal techniques to examine all of the mentioned networks

	Signaling networks	Gene regulatory networks	Metabolic networks
Boolean networks	+	++	
Bayesian networks	+	++	
Petri nets	++	+	++
Process algebras	++		
Constraint-based models	+	+	++
Differential equations	++	++	++
Rule-based models	++		
Interacting state machines	++		
Cellular automata	+	+	
Agent-based models	++		+

constraint-based models, and differential equations seem to be powerful enough to consider all aspects of metabolic system dynamics. A further observation indicates that Petri nets, constraint-based models, differential equations, and cellular automata are applied to all kinds of biological networks. This makes them potential candidates for whole-cell modeling. The most powerful technique is still differential equations modeling, which is also reflected by the data provided in the table. However, Petri nets are among the formalisms that cover most of the features to model all kinds of biological networks as described in Table 8.4. It is a universal graphical modeling concept for representing processes from different application fields in nearly all degrees of abstraction. Petri nets provide the qualitative modeling approach as well as the quantitative one. Furthermore, qualitative and quantitative formalism can be combined to one paradigm. The formalism is easy to understand and use.

Once a basic qualitative model is established, it can be successively enriched with quantitative data. Thus, parameter estimations based on experimentally derived data are not implicitly necessary in the network reconstruction process. Furthermore, models can be modeled discretely as well as continuously. It is even possible to integrate ODEs for precise model description.

Besides, Petri nets allow hierarchical structuring of models and thus offer the possibility of different detailed views for every observer of the model. Petri net theory provides a variety of established analysis techniques that are well suited and applicable to biological network modeling. Moreover, database information, as described in the following section, can be used to automatically reconstruct sophisticated network and Petri net models.

Table 8.4 Overview of implemented features for each modeling formalism based on [79]: (+) supported feature and (e) available through extension. Based on the provided data, the most powerful technique is the Petri net modeling as it includes the advantages and features of all other formalisms

	Visualization	Topology	Modularity	Hierarchy	Multistate	Compartments	Spatial	Qualitative	Synchronized	Stochastic	Continuous
Boolean networks	+	+						+	+	e	
Bayesian networks	+	+						+		+	
Petri nets	+	+	+	e	e			+	e	e	e
Process algebras			+	e		e		+		+	
Constraint-based models		+						+			
Differential equations							e			e	+
Rule-based models	+		+		+	+	e	+		+	+
Interacting state machines	+		+	+	+	+				+	
Cellular automata	+				+		+		+	+	
Agent-based models	+				+	+	+			+	

8.5 Network Reconstruction

A biological network, as described in Sect. 8.2, consists of a set of different biological elements being in interaction with each other. Such a network can be reconstructed by hand, with experimental data, information from literature, and/or database knowledge. In the first case, users need to put all involved elements into relation and draw the resulting models as a graph. They have several possibilities to model the system. They can use directed, undirected, mixed, or other graphs as presented in Sect. 8.3. Furthermore, they can use a standard graphical notation, such as SBGN for the visual modeling as presented in Sect. 8.4.13.

In terms of a network reconstruction with experimental data correlation, networks have to be reconstructed as described in Sect. 8.2. Therefore, a well-established modeling and analysis technique is necessary. One possible approach is the Bayesian networks as described in Sect. 8.4.9. Bayesian networks offer one way to automatically reconstruct signaling networks from experimentally derived data. The only disadvantage of this approach is the necessary input data. To be able to produce unambiguous results, a huge set of experimental data is mandatory.

A further way to reconstruct biological networks is by using text mining approaches [80, 81]. Text mining is equivalent to text analytics, with the goal of turning text into data for further analysis. This approach can be used, for example, to find interaction partners for a gene by analyzing a set of publications. The collected

data is then modeled as a graph. In general, this technique is based on statistical pattern learning. The main disadvantage of this approach is still the interpretation of the input text. In many cases relations are identified which are positive false or false positive. Although the analysis and results are becoming better and better, the resulting networks need to be evaluated by an expert.

A more reliable way to reconstruct biological networks is by querying biological databases. Therefore, more than 1,300 different biological databases exist that can be accessed. Using complex queries, data transformations, and data integration techniques, rudimentary data such as genes and proteins can be linked with each other. Many databases provide links between the different biological compounds. If such a link does not exist, it is even possible to establish connections by mining genomic databases. Hence, several attempts have been made to reconstruct metabolic pathways via genome sequence comparison [82, 83]. Such attempts have a certain limit, as the results do not reflect all involved molecular functions. Due to cellular functions, such as translation, transcription, post-modification, and many more processes with genome sequence comparison and analysis, it is often not possible to predict direct correlations and further regulatory or metabolic processes.

However, several databases do exist, which contain more detailed information about metabolic pathways, such as the KEGG database [21]. The information about the networks can be accessed via the Internet or by parsing provided flat-files. The disadvantage with online access is that the elements cannot be analyzed and combined with other -omic level data and experimental datasets. Therefore, flat files have to be processed, filtered, normalized, and integrated into one model. Actually, the KEGG database consists of more than 121 tables, where at least 23 tables are necessary to reconstruct the backbone of a biological network. The other tables store further information, such as diseases, drugs, and taxonomies (see Fig. 8.10 for a simplified scheme of the KEGG database structure). With access to that data, it is possible to reconstruct metabolic networks as they are presented by KEGG and to analyze the biological elements in detail or overall context.

In terms of biological network reconstruction using database information, each scientist should follow some basic recommendations:

1. All databases should be free of charge and accessible by using a SOAP or an API.
2. All databases should use the same terms, identifiers, and publication structures as cited in literature.
3. Provided datasets must be up to date and should not overlap.
4. The selected databases should be well curated.
5. Only databases which can be used for the reconstruction of biological networks should be integrated.
6. The used databases should be focusing on the mechanisms which should be modeled, such as metabolic pathways, signaling pathways, and protein-protein interaction networks.
7. It should be possible to query each integrated database separately or in combination with each other.

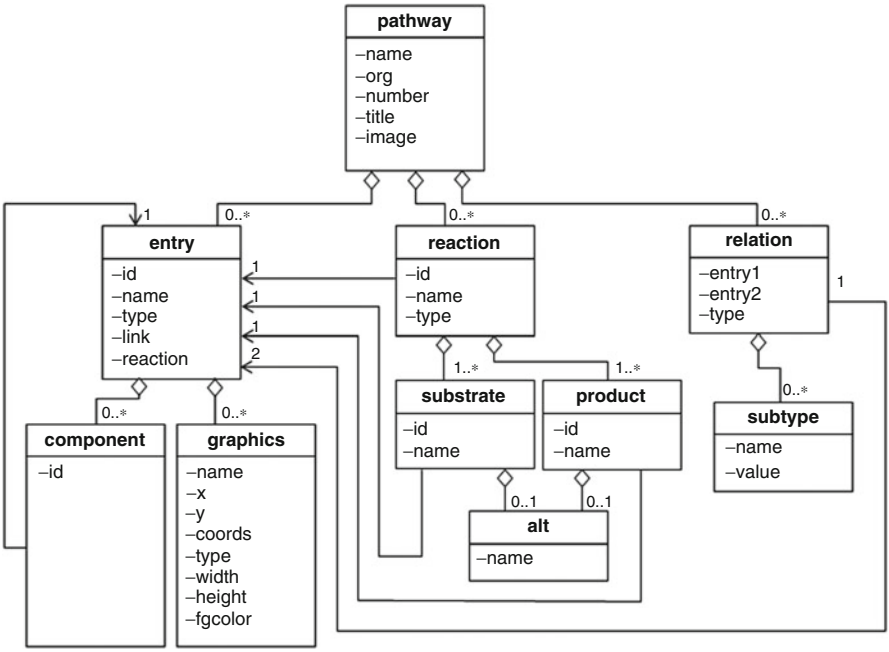

Fig. 8.10 Simplified scheme of the KEGG database structure [84]. The pathway element is the root element of the biological network, consisting of a list of entry, relation, and reaction elements. Theses entities specify the graph information. Additional elements specify more detailed information about the biological compounds, relations, and reactions within the model

8.6 Biological Network Exchange Formats

Molecular biotechnology, systems biology, bioinformatics, and many other disciplines in biology make it possible to reconstruct and analyze biological systems. More than 300 pathway or molecular interaction-related data resources, visualization, and analysis software tools have been developed.[1] However, the diversity of tools shows several problems in sharing and moving models between each other. An attempt to overcome this problem is the creation of standards [85–87].

In an online survey, Klipp et al. asked 125 researchers (75 % modelers, 4 % experimentalists, or 21 % both) covering various fields, such as modeling of individual pathways, investigation of complex processes, development and application of computational methods, and software development about their opinion on standards

[1]The number of software applications has been approximated by counting software tools that support SBML and CellML. Software tools are listed at http://www.sbml.org/ and http://www.cellml.org/

[88]. About 80 % of the scientists considered the creation of standards necessary or desirable. This is not surprising that science standards have many advantages as listed in the following:

- Model definitions and entities are based on ontologies, defined nomenclature, and restrictions. Thus, they become accessible and readable to a wide community.
- Standards improve communication between software tools, free exchange of information, and comparison between different studies, which results in more productive collaborations.
- Complementary resources from multiple simulation/analysis tools can work together, instead of redefining and reconstructing models in each tool.
- Reimplementation of models becomes easier or dispensable, which reduces duplication and redundancy.
- If tools are no longer supported, models developed within the tools can be still used if they are based on standards. Information, knowledge, and research progress is not lost and can be reused.
- Data curation teams can evaluate models without being restricted to a certain tool or formalism.
- In the publication process, any curator can process annotation and normalization before data is published and made available to the scientific community.

Scientists, simultaneously with both tool development and modeling projects, have developed standards to share, evaluate, and analyze knowledge and information. Standards are definitions in the form of common, inclusive, and computable languages. Here, only XML-based formats are considered, since it is used as universal language in data exchange. McEntire et al. [89] and Achard et al. [90] have shown in their studies that this language is very flexible and simple to use and, therefore, a powerful standard in bioinformatics and systems biology in comparison to Comma Separated Values (CSV), Excel, and other file formats. More than 85 standards can be found within systems biology [87].

For the modeling and sharing of biological models, main standards exist, such as the Systems Biology Ontology (SBO) [91], Systems Biology Markup Language (SBML) [92, 93], the CellML [94], and BioPAX [95]. For the graphical representation of biological pathways, languages such as the SBGN [77] have been introduced (see Sect. 8.4). Model description achieves human and computational usability, reusability, and interoperability when the encoded format is standardized. Models or software tools without standardization are only of limited use, as they do not provide the possibility to share, compare, and/or integrate large amount of systems. Thus, it is important to use common standards as described in the following section:

- **Systems Biology Ontology (SBO)**
  The SBO ontology [91] is a well-defined logic about biological terms, including single identifiers for each distinct entity, allowing clear reference and identification. Furthermore, it is augmented with terminological knowledge such as synonyms, abbreviations, and acronyms. The terminology is also used to

specify the type of the components being represented in a model and their role in systems biology descriptions. Thus, the ontology allows unambiguous and explicit understanding of the meaning of the involved components in a system and, moreover, enables mapping between elements of different models encoded in this format.

The ontology is a well-defined logic about biological terms, including a single identifier for each distinct entity, allowing clear reference and identification. It is composed of seven vocabulary branches: systems description parameter, participant role, modeling framework, mathematical expression, occurring entity representation, physical entity representation, and metadata representation. The terminology is also used to specify the type of components represented in a model and their role in systems biology descriptions. Thus, the ontology allows unambiguous and explicit understanding of the meaning of the involved components in a system and, moreover, enables mapping between elements of different models encoded in this format.

- **BioPAX**
  BioPAX is a standard language to represent biological pathways at the molecular and cellular level [95]. The main goal of BioPAX is the exchange of information between several pathway databases such as Reactome [96] and BioCyc [23]. It was introduced through a community process to make complete representation of basic cellular processes substantially easier to collect, to index, to interpret, and to share. BioPAX covers concepts such as metabolic and signaling pathways, gene regulatory networks, and genetic and molecular interactions. Therefore, it has a structure for substances, interactions, pathways, and links to organisms and experiments. The language is distributed as an ontology definition with associated documentation and a validator for checking. Therefore, the BioPAX community cooperates with the SBML and CellML mathematical modeling language communities. For better accessing and manipulating data in the BioPAX format, a house-implemented Java library called "Paxtool" is available. BioPAX Level 3 is currently available at http://www.biopax.org.
- **BioXSD**
  BioXSD is common exchange format for basic bioinformatics data [97]. Using this format, it should be possible to establish a common web service for the exchange of data for bioinformaticians in the World Wide Web. This format should fill gaps between specialized XML formats such as SBML [92, 93], MAGE-ML [98], GCDML [99], PDBML [100], MIF [101], and PhyloXML [102]. Therefore, BioXSD defines data formats such as biological sequences, sequence alignments, sequence annotation, and references to data, resources, and vocabularies in a variety of possibilities. BioXSD serves as a canonical data model and is available at http://bioxsd.org as version 1.1.
- **CellML**
  CellML [94, 103] is a language for representing mathematical models. Using differential algebraic equations, any cellular model can be represented in CellML. In addition, CellML represents entities using a component-based approach, where relationships between components are represented by connections. The

developers have implemented an API for working with CellML models and files. Thus, software developers do not need to reinvent the same functionality each time they develop a new tool. The API enables users to retrieve information, to manipulate, and to extend a model. The API interfaces are designed to be independent in any programming language, platform, or vendor. At the present time, CellML is available at http://www.cellml.org in version 1.1.

- **MathML**
  MathML is a low-level specification for describing mathematics [104, 105]. It is used wherever mathematics needs to be handled by software, such as mathematical expressions in web pages and workflows in science and technology. Actually, MathML is available at http://www.w3.org/Math/ as version 3.
- **PDBML**
  The PDB database is the single worldwide repository for macromolecular structure data [106]. For more than 30 years, the data resources have used a column-oriented format to store and share archival entries [100]. Facing more and more complex data for macromolecular structures, the used data format constrained several limitations such as internal structure and the organization of records. Therefore, a new XML-based data format, called PDBML, has been introduced [100]. It builds the content of the PDB exchange dictionary and can be used as a specific exchange medium for detailed molecular protein structures, such as data derived from experimental crystallography. PDBML is currently available at http://pdbml.pdb.org as version 3.3 to all users.
- **Systems Biology Markup Language (SBML)**
  SBML is an exchange format for representing biochemical reaction networks [92, 93]. Using SBML, users are able to describe models in many areas of computational biology, including cell signaling pathways, metabolic pathways, and gene regulation. Therefore, SBML has the structure, ontology, and links, for pathways and interactions. To enable mathematical descriptions, the SBML Level 2 uses MathML for more complex mathematical formulas. This extends the features of SBML and also results in a greater compatibility with CellML. Furthermore, it provides the possibility to specify delay functions and define discrete events that can occur at specified transitions in a certain state in biological models. In order to help users to read, write, manipulate, translate, and validate SBML files and data streams, the LibSBML API is available in different common programming languages, such as Java, C, and C++. Presently, SBML Level 2 is available at http://sbml.org/Software/libSBML and SBML Level 3 is being developed.

One of the main standards for the modeling of biological systems is the Systems Biology Ontology. Using this standard ensures the usability, reusability, and interoperability of biological models. Furthermore, data exchange standards can easily access models encoded in this format. For instance, SBML, MathML, and CellML support SBO definitions, which makes it easy to translate any kind of SBO model into such an exchange format. However, there is a significant difference in the scope of the mentioned standard exchange formats. By studying the most

important formats and considering recommendations from literature [86,87], SBML and CellML are proposed as a means for the exchange of biochemical reaction networks and models between different software tools. They provide an ontology and structure that can even be used for simulations. They also provide constructs that are similar to the object models used in packages specialized for simulating and analyzing biochemical networks. CellML and SBML, embedding MathML, provide users with the possibility for the representation of whole models in differential algebraic expressions. Besides, SBML and CellML have an API, which allows reading, writing, and manipulating models in an easy manner. Furthermore, SBML and CellML have much in common, since the development of both standards takes place cooperatively. Formats such as PDBML only focus on particular substances. Thus, they are not appropriate for network models. This also applies to MathML, which only provides basic mathematics. Furthermore, BioXSD and BioPAX exist and can be used as data standards. However, BioXSD is focused on data that is not supported by the main formats and thus very specialized and not capable of representing the entire biological systems. BioPAX is only focused on pathway maps, which can be shared between databases and tools. SBML and CellML can support dynamic systems in ways not possible for BioPax.

8.7 Where to Find Biological Databases and Tools for Network Reconstruction and Modeling

The first biological database emerged in 1965 when Margaret Dayhoff published the Atlas of Protein Sequence and Structure [107]. In the 1970s the first protein structure database, called PDB was found [108–110]. A few years later in 1981, the first repository for nucleotide sequences was established called EMBL [111, 112] and 1 year later the GenBank [113,114]. Since then, more and more biological databases have developed. The 19th annual database issue of NAR now lists more than 1,380 databases in molecular biology [115]. The Pathguide [116], a meta-database with an overview of more than 325 biological pathway-related resources, with more than 100 databases focused on protein-protein interaction, is an additional important resource for biological databases. To make it easier for researchers to quickly find relevant information about useful molecular resources, tools, and databases, community-curated databases with content and links to other biological databases were established. Some of the most important are MetaBase [117], OBRC [118], BioDBCore [119], and the Bioinformatics Links Directory [120, 121]. Currently, more than 1,800 entries are listed in MetaBase, each describing different biological databases. BioDBCore gives a brief description of the core attributes of biological databases, whereas OBRC contains annotations and links for more than 1,700 bioinformatics databases and software tools. The Bioinformatics Links Directory curates links to software tools and databases. Using these resources, users have the possibility to contribute, update, and maintain database content.

Concerning software tools in bioinformatics, in 2011, the SBML website[2] listed more than 200 software tools which provide biological modeling based on the SBML [92, 93]. Going further into details, Copeland et al. highlighted a small, representative portion of available tools from each -omic area [122]. Still, this review lists more than 30 tools specialized in biological modeling. However, the state of the-art applications CellDesigner [123], Cell Illustrator [124], Cytoscape [125], E-Cell [126], Gepasi [127, 128], JDesigner [129], VANESA in combination with the PNlib [130, 131], and Snoopy [132, 133] are able to model, reconstruct, visualize, and simulate biological systems in one single comprehensive framework.

8.7.1 CellDesigner

CellDesigner is a structured diagram editor for drawing gene regulatory and biochemical networks. It was developed by the Systems Biology Institute (SBI) in Tokyo, Japan [123]. The core members of this software application are Akira Funahashi, Hiroaki Kitano, and Akiya Jouraku. The main goal of this application is to visually represent biochemical reactions in a comprehensive graphical notation such as SBGN (Systems Biology Graphical Notation) [77]. Besides, in the new version it enables users to connect from species name or ID to the databases Saccharomyces Genome Database [134], iHOP (Information Hyperlinked over Proteins) [135], and the Genome Network Platform (http://genomenetwork.nig.ac.jp). Furthermore, it is possible to get basic information about a biological element from PubMed [136] or Entrez Gene, the search engine from NCBI (http://www.ncbi.nlm.nih.gov). To assist users in the simulation, CellDesigner is able to connect to the SBML ODE Solver [137] and Copasi, a biochemical network simulator [138]. Simulations can be set up in a control panel, where users are able to adjust system amounts and parameters. CellDesigner is free of charge and available at http://www.celldesigner.org in version 4.2 running under Windows and Linux.

8.7.2 Cell Illustrator

The software application Cell Illustrator [124] is a software platform for systems biology that uses the concept of the Petri net language for the modeling and simulating of biological networks. The first version of Cell Illustrator was published as Genomic Object Net [139] in 2000 under Matsuno et al. at the Faculty of Science, Yamaguchi University, Japan. The software application employs the concept of a hybrid Petri net as the modeling and simulation method. To handle any type of objects, the existing paradigm has been extended to hybrid functional Petri nets

[2] http://sbml.org/

with extension (HFPNe). This paradigm is more suitable for biological network modeling and simulation, since HFPNe can handle discrete and continuous events simultaneously. Any kind of function can be assigned to delay, weight, and speed parameters of these elements. Additionally, ordinary differential equations can be modeled and integrated into a subset of HFPNe.

Furthermore, Cell Illustrator is able to import pathways or single reactions from the TRANSPATH database [140]. To import networks from other tools, SBML, CellML, and BioPAX data exchange formats are supported. In addition, Cell Illustrator has its own format called CSML. Simulation results can be visualized in either 2D or 3D plots in an all-in-one-window environment. To make the network visualization more legible, graph grid layout algorithms are implemented. The latest version of Cell Illustrator is version 5.0, which is commercially an online version available at http://www.cellillustrator.com.

8.7.3 Cytoscape

Cytoscape is an open-source bioinformatics software platform for data integration and visualization [125]. The first version of Cytoscape was published by Shannon et al. from the Institute for Systems Biology, Seattle, Washington [141]. Nowadays, it is supported and funded by many different institutions, particularly by Agilent Technologies, University of Toronto, Institute Pasteur, Memorial Sloan-Kettering Cancer Center, Institute for Systems Biology, and the University of California San Diego. Primarily, Cytoscape enables users to visualize molecular interaction networks and biological pathways and integrate these with any type of attribute data, such as gene expression profiles. Furthermore, Cytoscape supports standard network and annotation files such as BioPAX [95], and SBML. Additional features are available as plugins, which are developed by third parties focusing on network and molecular profiling analyses, new layouts, additional file format support, scripting, and connection with databases. For network reconstruction there is the plug-in BioNetBuilder [142], which uses the databases KEGG [21], HPRD [143], BioGrid [144], and GO [145], among others for its modeling. Furthermore, simulation plug-ins exist, such as the SimBoolNet [146], for the simulation of Boolean networks or FERN for the stochastic simulation and evaluation of reaction networks [147]. Most of the plug-ins are available free of charge. Cytoscape uses an open API based on Java technology and version 2.8.3 is available at http://www.cytoscape.org.

8.7.4 E-Cell

The E-Cell project [126] is an international research project aimed at modeling and reconstructing biological phenomena in silico. The main goal of this software

application is to develop a dynamical cell with all its functions. It has been developed by Hashimoto et al. at the Institute for Advanced Biosciences, Keio University, Yokohama, Japan. The software platform allows precise whole-cell simulations with object-oriented modeling. Therefore, numerical integration methods are encapsulated into biologically related object classes. Virtually any integration algorithm can be used for simulation [148]. Thus, users have the possibility to define functions of proteins, protein-protein interactions, protein-DNA interactions, regulation of gene expressions, and other cellular cell processes with a set of functions rules. Therefore, hundreds of reaction rules are provided and available for simulation progress. E-Cell version 3 is freely available at http://www.e-cell.org and runs on several different platforms such as Microsoft Windows and Linux.

8.7.5 Gepasi

Gepasi is a software application for the modeling and simulating of biochemical systems [127, 128]. It has been developed by Pedro Mendes at the Department of Biological Sciences, University of Wales, Aberystwyth, UK. Gepasi uses mathematical formulas to transform biochemical properties into kinetic models. It provides a number of tools to fit data, to optimize any function of the model, and to perform metabolic control analysis and linear stability analysis. Sophisticated numerical algorithms realize simulation processes and analysis tasks. The simulation results can be plotted in 2D and 3D. Furthermore, the software application supports SBML 1.0 import and export. The latest version of Gepasi is 3.30 and freely available at http://www.gepasi.org. It only runs using Microsoft Windows.

8.7.6 JDesigner

JDesigner is a software application that enables users to draw a biochemical network, which can be exported to SBML for further processing [129]. The development of JDesigner was supported by the California Institute of Technology, Pasadena, California, and more recently by the KECK Institute of applied sciences, Claremont, California USA. JDesigner represents networks by using one notation for chemical species, which can be decorated with visual cues. This is also possible for reactions. Although it is a network design tool it also supports simulations. It has the ability to use JARNAC as a simulation server via the Systems Biology Workbench (SBW) [129] which is an open-source framework connecting heterogeneous software applications. JDesigner is an open-source project distributed under the LGPL license and available at http://sbw.kgi.edu/software/jdesigner.htm.

8.7.7 VANESA

VANESA is a modeling software for the automatic reconstruction and analysis of biological networks based on life-science database information [131, 149–153] and constantly developed at the Bielefeld University. VANESA is platform independent and available free of charge at www.vanesa.sf.net. Using VANESA, scientists are able to model any kind of biological processes and systems as biological networks. Scientists have the possibility to automatically reconstruct important biomedical systems with information from the databases KEGG, MINT, IntAct, HPRD, and BRENDA. Furthermore, users have the possibility to use graph theoretical approaches in VANESA to identify regulatory structures and significant actors within the modeled systems. These structures can then be further investigated in the Petri net environment PNlib for hypothesis generation and in silico experiments.

The PNlib is the powerful new state-of-the-art Petri net simulation library [130]. Proß et al. have developed the PNlib library using the Modelica language [154] at the Department of Engineering and Mathematics, University of Applied Sciences, Bielefeld, Germany. Modelica was developed and promoted by the Modelica Association since 1996 for modeling, simulation, and programming. Primarily it is focused on physical and technical systems and processes. Now, Modelica, embedding the PNlib, provides the possibility to simulate biological systems. VANESA and the PNlib are based on the xHPNbio formalism [131]. The mathematical modeling concept xHPNbio was specially developed for scientists, based on the demands of biological processes. The focus of this formalism is the processing of experimental data to gain usable new insights about biological systems.

8.7.8 Snoopy

Snoopy [132, 133] is a unifying Petri net framework to investigate biomolecular networks. It has been designed and implemented by Heiner et al. at the Brandenburg University of Technology at Cottbus, Germany. The simulation environment comprises a family of related Petri net classes, such as time Petri nets, stochastic Petri nets, continuous Petri nets, hybrid Petri nets, colored Petri nets, and extended Petri nets, among others. The mentioned classes enhance standard Petri nets in various ways to meet the demands of biological scientists. For example, the extended Petri nets are characterized by read arcs, inhibitor arcs, equal arcs, and reset arcs. Using these formalisms, scientists are able to reconstruct and simulate any kind of dynamic network. Larger networks can be hierarchically structured. If further demands on the supported Petri nets should arise, the software application can be extended by new properties and even by new Petri net classes. This is possible due

to the generic data structure of the software application. Furthermore, users are able to move between the qualitative, stochastic, and continuous modeling paradigms. However, this transformation from one paradigm into another is not possible without information loss.

Simulation results are visualized within a built-in animation environment. To be able to share results with other scientists and software applications, Snoopy offers SBML support with both import and export functions. Snoopy is available for all major operating systems, such as Windows, Linux, and Mac OS-X. It is available free of charge at http://www-dssz.informatik.tu-cottbus.de/snoopy.html.

8.8 Discussion

Cellular life is very complex and governed by thousands of macroscopic functions being constantly carried out. To produce good theoretical models which can be used for hypothesis testing, the models need to be manageable. This can only be achieved by reducing a biological system to the known and essential parts, which are necessary to answer the underlying research questions. By trying to model a complete system, regardless of the lack of data and parameters, it is very likely that the modeled systems can be misleading. Therefore, any model needs to have a clear focus rather than model all levels of biological details.

One of the best ways to start modeling a biological system is by using biological networks. A small network consisting of known and already analyzed elements can be the initial point for the reconstruction of a more significant system. Therefore, there are different biological networks which can be used as powerful integrated frameworks to present, integrate, and visualize knowledge. As these networks are intuitive and easy to extend in knowledge, any scientist can work with them. With biological networks different -omic levels can be modeled, describing elements such as genes, RNAs, proteins, and metabolites being in interactions and relationships with each other. Moreover, biological databases can be used to reconstruct or enrich those networks with relevant information and new data. Kinetics and other information can be queried to model a system in a more precise way. With database integration modules, it is even possible to query multiple databases with one view instead of consulting each database separately. Besides, data integration tools filter, normalize, and link heterogeneous data from different distributed data sources.

A further advantage of biological networks is that a wide range of graphical theoretical analysis techniques can be applied on reconstructed models. Graph theory can give important clues about topological network properties, such as the identification of the most important nodes within a system, or average path lengths between different elements in a biological model. This is important in as much as biological networks can become large and complex. Scientists need a tool which assists them in identifying relevant information.

When it comes to simulating cell behavior, scientists often speak about ODE modeling. Indeed, it is one of the most powerful approaches, but needs prior

knowledge in mathematics and a complete set of biological data and parameters. These are high requirements for a modeling approach when scientists try to reconstruct and understand system behavior or unknown regulatory processes. Thus, a more intuitive approach is necessary, which can be used in the beginning without biological data and is still able to imitate and predict cell behavior. Therefore, Petri nets can be used for the description, simulation, and analysis of complex and distributed systems. Petri nets cover most of the needed features for network modeling and provide qualitative as well as quantitative modeling features. Furthermore, it is possible to integrate ODEs for precise model descriptions. Another advantage of these modeling techniques is that each result can be shared within the scientific community using data exchange formats.

References

1. Alberts B, Johnson A, Lewis J, Raff M, Roberts K, Walter P (2007) Molecular biology of the cell, 5th edn. Garland Science, New York
2. Eigen M (1971) Self organization of matter and the evolution of biological macromolecules. Naturwissenschaften 58(10):465–523
3. Schrödinger E (1955) What is life? The physical aspect of the living cell. The University Press, Cambridge, pp 1–12
4. Dronamraju KR, Schrödinger E (1999) Erwin Schrödinger and the origins of molecular biology. Genetics 153(3):1071–1076
5. Holum JR (1998) Fundamentals of general, organic, and biological chemistry, 6th edn. Wiley, New York
6. Crick FHC (1958) Ideas on protein synthesis. Symp Soc Exp Biol XII:1–2
7. Crick FHC (1970) Central dogma of molecular biology. Nature 227(5258):561–563
8. Alon U (2006) An introduction to systems biology: design principles of biological circuits. Simplicity in biology, 1st edn. Chapman and Hall/CRC, Boca Raton
9. Junker BH, Schreiber F (2011) Analysis of biological networks, 1st edn. Wiley-Interscience, Hoboken
10. De Las Rivas J, Fontanillo C (2010) Protein-protein interactions essentials: key concepts to building and analyzing interactome networks. PLoS Comput Biol 6(6):1–8
11. Waksman G (2005) Proteomics and protein-protein interactions. Volume 3 of biology, chemistry, bioinformatics, and drug design. Springer, New York, pp 50–89
12. Panchenko A, Przytycka T (2008) Protein-protein interactions and networks. Identification, computer analysis, and prediction, 1st edn. Springer, New York Incorporated, pp 1–53
13. Suthram S (2008) Dissertation: understanding cellular function through the analysis of protein interaction networks. ProQuest, University of California, San Diego, 191p
14. Kepes F (2007) Biological networks, 1st edn. World Scientific, Singapore
15. Zhang A (2009) Protein interaction networks: computational analysis, 1st edn. Cambridge University Press, Cambridge/New York, pp. 1–62
16. Klussmann E, Scott J, Aandahl EM (2008) Protein-protein interactions as new drug targets, 1st edn. Springer, Berlin
17. Stelzl U, Worm U, Lalowski M, Haenig C, Brembeck FH, Goehler H, Stroedicke M, Zenkner M, Schoenherr A, Koeppen S, Timm J, Mintzlaff S, Abraham C, Bock N, Kietzmann S, Goedde A, Toksöz E, Droege A, Krobitsch S, Korn B, Birchmeier W, Lehrach H, Wanker EE (2005) A human protein-protein interaction network: a resource for annotating the proteome. Cell 122(6):957–968

18. Altman A (2002) Signal transduction pathways in autoimmunity, vol 5. Karger, Basel/New York
19. Cheong R, Hoffmann A, Levchenko A (2008) Understanding NF-$kappa$B signaling via mathematical modeling. Mol Syst Biol 4:192–196
20. Kearns JD (2006) IκB provides negative feedback to control NF-κB oscillations, signaling dynamics, and inflammatory gene expression. J Cell Biol 173(5):659–664
21. Kanehisa M, Goto S, Sato Y, Furumichi M, Tanabe M (2012) KEGG for integration and interpretation of large-scale molecular data sets. Nucleic Acids Res 40(Database Issue):109–114
22. Keseler IM, Bonavides-Martinez C, Collado-Vides J, Gama-Castro S, Gunsalus RP, Johnson DA, Krummenacker M, Nolan LM, Paley S, Paulsen IT, Peralta-Gil M, Santos-Zavaleta A, Shearer AG, Karp PD (2009) EcoCyc: a comprehensive view of Escherichia coli biology. Nucleic Acids Res 37(Database Issue):464–470
23. Karp PD, Ouzounis CA, Moore-Kochlacs C, Goldovsky L, Kaipa P, Ahrén D, Tsoka S, Darzentas N, Kunin V, López-Bigas N (2005) Expansion of the BioCyc collection of pathway/genome databases to 160 genomes. Nucleic Acids Res 33(19):6083–6089
24. Hopfield JJ (1982) Neural networks and physical systems with emergent collective computational abilities. Proc Natl Acad Sci 79(8):2554–2558
25. Van Pelt J, Kamermans U, Levelt CN (2004) Development, dynamics and pathology of neuronal networks: from molecules to functional circuits. Elsevier Science and Technology, Amsterdam/Boston
26. Kunin V, Goldovsky L, Darzentas N, Ouzounis CA (2005) The net of life: reconstructing the microbial phylogenetic network. Genome Res 15(7):954–959
27. Gray SH (2008) Food webs: interconnecting food chains. Exploring science. Compass Point Books, Minneapolis, pp 1–48
28. De Ruiter PC, Wolters V, Moore JC (2005) Dynamic food webs. Multispecies assemblages, ecosystem development, and environmental change. Academic, Amsterdam/Boston, pp 3–10
29. Pimm SL (2002) Food webs. University of Chicago Press, Chicago, pp 1–34
30. Hoppensteadt FC (1982) Mathematical methods of population biology. Cambridge University Press, Cambridge/New York, pp 1–28
31. Diestel R (2000) Graphentheorie, vol 2. Springer, Berlin, pp 1–25
32. Pavlopoulos GA, Secrier M, Moschopoulos CN, Soldatos TG, Kossida S, Aerts J, Schneider R, Bagos PG (2011) Using graph theory to analyze biological networks. BioData Min 4:10
33. Koschützki D, Lehmann KA, Peeters L, Richter S, Tenfelde-Podehl D, Zlotowski O (2005) Network analysis: centrality indices. Lecture notes in computer science, vol 3418. Springer, Berlin, Heidelberg
34. Jeong H, Mason SP, Barabási A, Oltvai ZN (2001) Lethality and centrality in protein networks. Nature 411(6833):41–42
35. Hahn MW, Kern AD (2005) Comparative genomics of centrality and essentiality in three eukaryotic protein-interaction networks. Mol Biol Evol 22(4):803–806
36. Hage P, Harary F (1995) Eccentricity and centrality in networks. Soc Netw 17:57–63
37. Sabidussi G (1966) The centrality index of a graph. Psychometrika 31(4):581–603
38. Freeman LC (1977) A set of measures of centrality based on betweenness. Sociometry 40:35–41
39. Bonacich P (1972) Factoring and weighting approaches to status scores and clique identification. J Math Sociol 2:113–120
40. Lewinski M (2012) Mathematischer Netzwerkvergleich anhand multipler Netzwerk Charakteristika. Master thesis at the Bielefeld University, pp 58–70
41. Stein ML, Stein PR (1967) Enumeration of linear graphs and connected linear graphs up to p = 18 Points. Report LA-3775, Los Alamos Scientific Laboratory of the University of California
42. van den Berg H (2011) Mathematical models of biological systems (Oxford biology), 1st edn. Oxford University Press, Oxford
43. Tyson J, Chen KC, Novak B (2003) Sniffers, buzzers, toggles and blinkers: dynamics of regulatory and signaling pathways in the cell. Curr Opin Cell Biol 15(2):221–231

44. Gizzatkulov M, Goryanin I, Metelkin EA, Mogilevskaya EA, Peskov KV, Demin OV (2010) DBSolve Optimum: a software package for kinetic modeling which allows dynamic visualization of simulation results. BMC Syst Biol 4:109
45. Demin O, Goryanin I (2008) Kinetic modelling in systems biology. Chapman & Hall/CRC, Boca Raton
46. Rumbaugh J, Eddy F (1991) Object-oriented modeling and design. Prentice-Hall, Englewood Cliffs, pp 15–57
47. Johnson CG, Goldman JP, William JG (2004) Simulating complex intracellular processes using object-oriented computational modelling. Prog Biophys Mol Biol 86(3):379–406
48. Francesca C, Daniele M, Marco G (2009) Modeling biological pathways: an object-oriented like methodology based on mean field analysis. Trans IRE 1:117–122
49. Dorkeld F, Perrière G, Gautier C (1993) Object-oriented modelling in molecular biology. In: Ganascia JG (ed) Proceedings of the artificial intelligence and genome workshop, IJCAI, pp 99–106
50. Lee M, Starr LLJ (1994) Object-oriented analysis in the real-world. Embed Syst Program 7(6):24–37
51. Kurth W (2007) Specification of morphological models with L-systems and relational growth grammars. J Interdiscip Image Sci 5:1–25
52. Prusinkiewicz P, Lindenmayer A (1990) The algorithmic beauty of plants. Springer, New York, pp 1–46
53. Orth JD, Conrad TM, Na J, Lerman JA, Nam H, Feist AM, Palsson BO (2011) A comprehensive genome-scale reconstruction of Escherichia coli metabolism. Mol Syst Biol 7:1–9
54. Kauffman KJ, Prakash P, Edwards JS (2003) Advances in flux balance analysis. Curr Opin Biotechnol 14(5):491–496
55. Wiechert W (2001) 13C metabolic flux analysis. Metab Eng 3(3):195–206
56. Efroni S, Harel D, Cohen IR (2003) Toward rigorous comprehension of biological complexity: modeling, execution, and visualization of thymic T-cell maturation. Genome Res 13(11):2485–2497
57. Kam N, Cohen IR, Harel D (2001) Proceedings IEEE symposia on human-centric computing languages and environments (Cat. No.01TH8587). In: HCC 2001. IEEE symposium on human-centric computing languages and environments, vol 1, Auckland. IEEE, pp 15–22
58. Danos V, Laneve C (2004) Formal molecular biology. Theor Comput Sci 325:69–110
59. Ermentrout GB, Edelsteinkeshet L (1993) Cellular automata approaches to biological modeling. J Theor Biol 160:97–133
60. Wishart DS, Yang R, Arndt D, Tang P, Cruz J (2005) Dynamic cellular automata: an alternative approach to cellular simulation. In Silico Biol 5(2):139–161
61. Wurthner JU, Mukhopadhyay AK, Peimann CJ (1999) A cellular automaton model of cellular signal transduction. Comput Biol Med 30:1–21
62. Gardner M (1970) Mathematical games: the fantastic combinations of John Conway's new solitaire game "life". Sci Am 1:120–123
63. Zhang L, Athale CA, Deisboeck TS (2007) Development of a three-dimensional multiscale agent-based tumor model: simulating gene-protein interaction profiles, cell phenotypes and multicellular patterns in brain cancer. J Theor Biol 244:96–107
64. Grant MR, Mostov KE, Tlsty TD, Hunt CA (2006) Simulating properties of in vitro epithelial cell morphogenesis. PLoS Comput Biol 2(10):129–134
65. Li NYK, Verdolini K, Clermont G, Mi Q, Rubinstein EN, Hebda PA, Vodovotz Y (2008) A Patient-Specific in silico model of Inflammation and Healing Tested in Acute Vocal Fold Injury. PLoS ONE 3(7):1–11
66. Lee S, Dudley AM, Drubin D, Silver PA, Krogan NJ, Pe'er D, Koller D (2009) Learning a Prior on Regulatory Potential from eQTL Data. PLoS Genet 5:1–24
67. Pe'er D (2005) Bayesian network analysis of signaling networks: a primer. Trans IRE Prof Group Audio 2005(281):14–24

68. Segal E, Shapira M, Regev A, Pe'er D, Botstein D, Koller D, Friedman N (2003) Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. Nat Genet 34(2):166–176
69. Kauffman SA (1969) Metabolic stability and epigenesis in randomly constructed genetic nets. J Theor Biol 22(3):437–467
70. Gershenson C (2004) Introduction to random boolean networks, pp 1–14. eprint arXiv nlin/0408006
71. Thomas R (1973) Boolean formalization of genetic control circuits. J Theor Biol 42(3):563–585
72. Thomas R, D'Ari RTR (1990) Biological feedback. CRC, Boca Raton, pp 1–316
73. Bernot G, Cassez F, Comet JP, Delaplace F, Müller C, Roux O (2007) Semantics of biological regulatory networks. Electron Notes Theor Comput Sci 180(3):3–14
74. Petri CA (1962) Dissertation: Kommunikation mit Automaten. Schriften des Rheinisch-Westfälischen Institutes für Instrumentelle Mathematik an der Universität Bonn
75. Reisig W (1992) A primer in Petri net design, 1st edn. Springer, Berlin/New York
76. Petri C, Reisig W (2008) Petri net. Scholarpedia 3(4):6477
77. Le Novère N, Hucka M, Mi H, Moodie S, Schreiber F, Sorokin A, Demir E, Wegner K, Aladjem MI, Wimalaratne SM, Bergman FT, Gauges R, Ghazal P, Kawaji H, Li L, Matsuoka Y, Villeger A, Boyd SE, Calzone L, Courtot M, Dogrusoz U, Freeman TC, Funahashi A, Ghosh S, Jouraku A, Kim S, Kolpakov F, Luna A, Sahle S, Schmidt E, Watterson S, Wu G, Goryanin I, Kell DB, Sander C, Sauro H, Snoep JL, Kohn K, Kitano H (2009) The systems biology graphical notation. Nat Biotechnol 27(8):735–741
78. SBGN Documentation: SBGN entity relationship diagram. http://www.sbgn.org/Documents/ER_L1_Examples. Online; accessed 17 Aug 2013
79. Machado D, Costa RS, Rocha M, Ferreira EC, Tidor B, Rocha I (2011) Modeling formalisms in Systems Biology. AMB Express 1:45–59
80. Krallinger M, Alonso-Allende Erhardt M, Valencia A (2005) Text-mining approaches in molecular biology and biomedicine. Drug Disc Today 10(6):439–445
81. Hoffmann R, Krallinger M, Andres E, Tamames J, Blaschke C, Valencia A (2005) Text mining for metabolic pathways, signaling cascades, and protein networks. Signal Transduct Knowl Environ 1(283):1–21
82. Mushegian AR, Koonin EV (1996) A minimal gene set for cellular life derived by comparison of complete bacterial genomes. Proc Natl Acad Sci 93(19):10268–10273
83. Bono H, Ogata H, Goto S, Kanehisa M (1998) Reconstruction of amino acid biosynthesis pathways from the complete genome sequence. Genome Res 8:203–210
84. KEGG Documentation: Overview of KGML. http://www.kegg.jp/kegg/xml/docs/. Online; accessed 17 Aug 2013
85. Milanesi L, Romano P, Castellani G, Remondini D, Lio P (2009) Trends in modeling Biomedical Complex Systems. BMC Bioinform 10(12):1–13
86. Sauro HM, Bergmann FT (2008) Standards and ontologies in computational systems biology. Essays Biochem 45:211–222
87. Strömbäck L, Hall D, Lambrix P (2007) A review of standards for data exchange within systems biology. Proteomics 7(6):857–867
88. Klipp E, Liebermeister W, Helbig A, Kowald A, Schaber J (2007) Systems biology standards–the community speaks. Nat Biotechnol 25(4):390–391
89. McEntire R, Karp P, Abernethy N, Benton D, Helt G, DeJongh M, Kent R, Kosky A, Lewis S, Hodnett D, Neumann E, Olken F, Pathak D, Tarczy-Hornoch P, Toldo L, Topaloglou T (2000) An evaluation of ontology exchange languages for bioinformatics. Int Conf Intell Syst Mol Biol 8:239–250
90. Achard F, Vaysseix G, Barillot E (2001) XML, bioinformatics and data integration. Bioinformatics 17(2):115–125
91. Courtot M, Juty N, Knüpfer C, Waltemath D, Zhukova A, Dräger A, Dumontier M, Finney A, Golebiewski M, Hastings J, Hoops S, Keating S, Kell DB, Kerrien S, Lawson J, Lister A, Lu J, Machne R, Mendes P, Pocock M, Rodriguez N, Villeger A, Wilkinson DJ, Wimalaratne

S, Laibe C, Hucka M, Le Novère N (2011) Controlled vocabularies and semantics in systems biology. Mol Syst Biol 7:1–12
92. Finney A, Hucka M (2003) Systems biology markup language: Level 2 and beyond. Biochem Soc Trans 31:1472–1473
93. Hucka M, Finney A, Sauro HM, Bolouri H, Doyle JC, Kitano H, Arkin AP, Bornstein BJ, Bray D, Cornish-Bowden A, Cuellar AA, Dronov S, Gilles ED, Ginkel M, Gor V, Goryanin II, Hedley WJ, Hodgman TC, Hofmeyr JH, Hunter PJ, Juty NS, Kasberger JL, Kremling A, Kummer U, Le Novere N, Loew LM, Lucio D, Mendes P, Minch E, Mjolsness ED, Nakayama Y, Nelson MR, Nielsen PF, Sakurada T, Schaff JC, Shapiro BE, Shimizu TS, Spence HD, Stelling J, Takahashi K, Tomita M, Wagner J, Wang J (2003) The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. Bioinformatics 19(4):524–531
94. Miller AK, Marsh J, Reeve A, Garny A, Britten R, Halstead M, Cooper J, Nickerson DP, Nielsen PF (2010) An overview of the CellML API and its implementation. BMC Bioinform 11:178–180
95. Demir E, Cary MP, Paley S, Fukuda K, Lemer C, Vastrik I, Wu G, D'Eustachio P, Schaefer C, Luciano J, Schacherer F, Martinez-Flores I, Hu Z, Jimenez-Jacinto V, Joshi-Tope G, Kandasamy K, Lopez-Fuentes AC, Mi H, Pichler E, Rodchenkov I, Splendiani A, Tkachev S, Zucker J, Gopinath G, Rajasimha H, Ramakrishnan R, Shah I, Syed M, Anwar N, Babur Ö, Blinov M, Brauner E, Corwin D, Donaldson S, Gibbons F, Goldberg R, Hornbeck P, Luna A, Murray-Rust P, Neumann E, Ruebenacker O, Reubenacker O, Samwald M, van Iersel M, Wimalaratne S, Allen K, Braun B, Whirl-Carrillo M, Cheung KH, Dahlquist K, Finney A, Gillespie M, Glass E, Gong L, Haw R, Honig M, Hubaut O, Kane D, Krupa S, Kutmon M, Leonard J, Marks D, Merberg D, Petri V, Pico A, Ravenscroft D, Ren L, Shah N, Sunshine M, Tang R, Whaley R, Letovksy S, Buetow KH, Rzhetsky A, Schachter V, Sobral BS, Dogrusoz U, McWeeney S, Aladjem M, Birney E, Collado-Vides J, Goto S, Hucka M, Le Novère N, Maltsev N, Pandey A, Thomas P, Wingender E, Karp PD, Sander C, Bader GD (2010) The BioPAX community standard for pathway data sharing. Nat Biotechnol 28(9):935–942
96. Matthews L, Gopinath G, Gillespie M, Caudy M, Croft D, de Bono B, Garapati P, Hemish J, Hermjakob H, Jassal B, Kanapin A, Lewis S, Mahajan S, May B, Schmidt E, Vastrik I, Wu G, Birney E, Stein L, D'Eustachio P (2009) Reactome knowledgebase of human biological pathways and processes. Nucleic Acids Res 37(Database Issue):619–622
97. Kalas M, Puntervoll P, Joseph A, Bartaseviciute E, Topfer A, Venkataraman P, Pettifer S, Bryne JC, Ison J, Blanchet C, Rapacki K, Jonassen I (2010) BioXSD: the common data-exchange format for everyday bioinformatics web services. Bioinformatics 26(18):540–546
98. Spellman PT, Miller M, Stewart J, Troup C, Sarkans U, Chervitz S, Bernhart D, Sherlock G, Ball C, Lepage M, Swiatek M, Marks WL, Goncalves J, Markel S, Iordan D, Shojatalab M, Pizarro A, White J, Hubley R, Deutsch E, Senger M, Aronow BJ, Robinson A, Bassett D, Stoeckert CJ, Brazma A (2002) Design and implementation of microarray gene expression markup language (MAGE-ML). Genome Biol 3(9):1–9
99. Kottmann R, Gray T, Murphy S, Kagan L, Kravitz S, Lombardot T, Field D, Glöckner FO (2008) A standard MIGS/MIMS compliant XML Schema: toward the development of the Genomic Contextual Data Markup Language (GCDML). OMICS 12(2):115–121
100. Westbrook J, Ito N, Nakamura H, Henrick K, Berman HM (2005) PDBML: the representation of archival macromolecular structure data in XML. Bioinformatics 21(7):988–992
101. Hermjakob H, Montecchi-Palazzi L, Bader G, Wojcik J, Salwinski L, Ceol A, Moore S, Orchard S, Sarkans U, von Mering C, Roechert B, Poux S, Jung E, Mersch H, Kersey P, Lappe M, Li Y, Zeng R, Rana D, Nikolski M, Husi H, Brun C, Shanker K, Grant SG, Sander C, Bork P, Zhu W, Pandey A, Brazma A, Jacq B, Vidal M, Sherman D, Legrain P, Cesareni G, Xenarios I, Eisenberg D, Steipe B, Hogue C, Apweiler R (2004) The HUPO PSI's molecular interaction format–a community standard for the representation of protein interaction data. Nat Biotechnol 22(2):177–183
102. Han MV, Zmasek CM (2009) phyloXML: XML for evolutionary biology and comparative genomics. BMC Bioinform 10:356

103. Cuellar AA, Lloyd CM, Nielsen PF, Bullivant DP, Nickerson DP, Hunter PJ (2003) An Overview of CellML 1.1, a biological model description language. Simulation 79(12):740–747
104. Sandhu P (2003) The MathML handbook. Charles River Media, Hingham, pp 1–10
105. Ausbrooks R, Buswell S, Carlisle D, Chavchanidze G, Dalmas S, Devitt S, Diaz A, Dooley S, Hunter R, Ion P, Kohlhase M, Lazrek A, Libbrecht P, Miller B, Miner R, Rowley C, Sargent M, Smith B, Soiffer N, Sutor R, Watt S (2010) Mathematical Markup Language (MathML) Version 3.0, pp 115–121. W3C recommendations-www.w3.org
106. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE (2000) The Protein Data Bank. Nucleic Acids Res 28:235–242
107. Strasser BJ, Dayhoff MO (2010) Collecting, comparing, and computing sequences: the making of Margaret O. Dayhoff's Atlas of Protein Sequence and Structure, 1954–1965. J Hist Biol 43(4):623–660
108. Bernstein FC, Koetzle TF, Williams GJ, Meyer EF, Brice MD, Rodgers JR, Kennard O, Shimanouchi T, Tasumi M (1977) The Protein Data Bank: a computer-based archival file for macromolecular structures. J Mol Biol 112(3):535–542
109. Meyer EF (1997) The first years of the Protein Data Bank. Protein Sci 6(7):1591–1597
110. Berman H, Henrick K, Nakamura H, Markley JL (2007) The worldwide Protein Data Bank (wwPDB): ensuring a single, uniform archive of PDB data. Nucleic Acids Res 35(Database Issue):301–303
111. Hamm GH, Cameron GN (1986) The EMBL data library. Nucleic Acids Res 14:5–9
112. Cochrane G, Akhtar R, Bonfield J, Bower L, Demiralp F, Faruque N, Gibson R, Hoad G, Hubbard T, Hunter C, Jang M, Juhos S, Leinonen R, Leonard S, Lin Q, Lopez R, Lorenc D, McWilliam H, Mukherjee G, Plaister S, Radhakrishnan R, Robinson S, Sobhany S, Hoopen PT, Vaughan R, Zalunin V, Birney E (2009) Petabyte-scale innovations at the European Nucleotide Archive. Nucleic Acids Res 37(Database Issue):19–25
113. Burks C, Fickett JW, Goad WB, Kanehisa M, Lewitter FI, Rindone WP, Swindell CD, Tung CS, Bilofsky HS (1985) The GenBank nucleic acid sequence database. Comput Appl Biosci 1(4):225–233
114. Benson DA, Karsch-Mizrachi I, Clark K, Lipman DJ, Ostell J, Sayers EW (2012) GenBank. Nucleic Acids Res 40(Database Issue):48–53
115. Galperin MY, Fernandez-Suarez XM (2012) The 2012 Nucleic Acids Research Database Issue and the online Molecular Biology Database Collection. Nucleic Acids Res 40(Database Issue):1–8
116. Bader GD, Cary MP, Sander C (2006) Pathguide: a pathway resource list. Nucleic Acids Res 34(Database Issue):504–506
117. Bolser DM, Chibon PY, Palopoli N, Gong S, Jacob D, Del Angel VD, Swan D, Bassi S, Gonzalez V, Suravajhala P, Hwang S, Romano P, Edwards R, Bishop B, Eargle J, Shtatland T, Provart NJ, Clements D, Renfro DP, Bhak D, Bhak J (2012) MetaBase–the wiki-database of biological databases. Nucleic Acids Res 40(Database Issue):1250–1254
118. Chen YB, Chattopadhyay A, Bergen P, Gadd C, Tannery N (2007) The Online Bioinformatics Resources Collection at the University of Pittsburgh Health Sciences Library System – a one-stop gateway to online bioinformatics databases and software tools. Nucleic Acids Res 35(Database Issue):780–785
119. Bateman A (2010) Curators of the world unite: the International Society of Biocuration. Bioinformatics 26(8):991
120. Fox JA, Butland SL, McMillan S, Campbell G, Ouellette BF (2005) The Bioinformatics Links Directory: a compilation of molecular biology web servers. Nucleic Acids Res 33(Web Server issue):3–24
121. Brazas MD, Yim DS, Yamada JT, Ouellette BF (2011) The 2011 Bioinformatics Links Directory update: more resources, tools and databases and features to empower the bioinformatics community. Nucleic Acids Res 39(Web Server issue):3–7
122. Copeland WB, Bartley BA, Chandran D, Galdzicki M, Kim KH, Sleight SC, Maranas CD, Sauro HM (2012) Computational tools for metabolic engineering. Metab Eng 14(3):270–280

123. Funahashi A, Morohashi M, Kitano H, Tanimura N (2003) CellDesigner: a process diagram editor for gene-regulatory and biochemical networks. Biosilico 1(5):159–162
124. Nagasaki M, Saito A, Jeong E, Li C, Kojima K, Ikeda E, Miyano S (2010) Cell Illustrator 4.0: a computational platform for systems biology. In Silico Biol 10:5–26
125. Smoot ME, Ono K, Ruscheinski J, Wang PL, Ideker T (2011) Cytoscape 2.8: new features for data integration and network visualization. Bioinformatics 27(3):431–432
126. Tomita M, Hashimoto K, Takahashi K, Shimizu TS, Matsuzaki Y, Miyoshi F, Saito K, Tanida S, Yugi K, Venter JC (1999) E-CELL: software environment for whole-cell simulation. Bioinformatics 15:72–84
127. Mendes P (1997) Biochemistry by numbers: simulation of biochemical pathways with Gepasi 3. Trends Biochem Sci 22(9):361–363
128. Mendes P (1993) GEPASI: a software package for modelling the dynamics, steady states and control of biochemical and other systems. Comput Appl Biosci 9(5):563–571
129. Sauro HM, Hucka M, Finney A, Wellock C, Bolouri H, Doyle J, Kitano H (2002) Next generation simulation tools: the Systems Biology Workbench and BioSPICE integration. OMICS: J Integr Biol 7(4):355–372
130. Proß S, Bachmann B (2011) An advanced environment for hybrid modeling of biological systems based on Modelica. J Integr Bioinform 8:152
131. Proß S, Janowski SJ, Bachmann B, Kaltschmidt C, Kaltschmidt B (2012) PNlib- a Modelica library for simulation of biological systems based on extended hybrid Petri nets. In: Proceedings of the 3rd international workshop on biological processes and Petri nets, vol 852, pp 1–16
132. Heiner M, Richter R, Schwarick M, Rohr C (2008) Snoopy-A tool to design and execute graph-based formalisms. Petri Net Newsl 74:8–22
133. Rohr C, Marwan W, Heiner M (2010) Snoopy- a unifying Petri net framework to investigate biomolecular networks. Bioinformatics 26(7):974–975
134. Cherry JM, Hong EL, Amundsen C, Balakrishnan R, Binkley G, Chan ET, Christie KR, Costanzo MC, Dwight SS, Engel SR, Fisk DG, Hirschman JE, Hitz BC, Karra K, Krieger CJ, Miyasato SR, Nash RS, Park J, Skrzypek MS, Simison M, Weng S, Wong ED (2012) Saccharomyces Genome Database: the genomics resource of budding yeast. Nucleic Acids Res 40(Database Issue):700–705
135. Hoffmann R, Valencia A (2004) A gene network for navigating the literature. Nat Genet 36(7):664–664
136. Wheeler DL, Church DM, Edgar R, Federhen S, Helmberg W, Madden TL, Pontius JU, Schuler GD, Schriml LM, Sequeira E, Suzek TO, Tatusova TA, Wagner L (2004) Database resources of the National Center for Biotechnology Information: update. Nucleic Acids Res 32(Database Issue):35–40
137. Machné R, Finney A, Müller S, Lu J, Widder S, Flamm C (2006) The SBML ODE Solver Library: a native API for symbolic and fast numerical analysis of reaction networks. Bioinformatics 22(11):1406–1407
138. Hoops S, Sahle S, Gauges R, Lee C, Pahle J, Simus N, Singhal M, Xu L, Mendes P, Kummer U (2006) COPASI – a COmplex PAthway SImulator. Bioinformatics 22(24):3067–3074
139. Matsuno H, Doi A, Drath R, Miyano S (2000) Genomic object net: object oriented representation of biological systems. Genome Inform Ser 11:229–230
140. Krull M, Pistor S, Voss N, Kel A, Reuter I, Kronenberg D, Michael H, Schwarzer K, Potapov A, Choi C, Kel-Margoulis O, Wingender E (2006) TRANSPATH: an information resource for storing and visualizing signaling pathways and their pathological aberrations. Nucleic Acids Res 34(Database Issue):546–551
141. Shannon P, Andrew M, Ozier O, Baliga NS, Wang JT, Ramage D, Amin N, Schwikowski B, Ideker T (2003) Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome Res 13(11):2498–2504
142. Avila-Campillo I, Drew K, Lin J, Reiss DJ, Bonneau R (2007) BioNetBuilder: automatic integration of biological networks. Bioinformatics 23(3):392–393

143. Keshava Prasad TS, Goel R, Kandasamy K, Keerthikumar S, Kumar S, Mathivanan S, Telikicherla D, Raju R, Shafreen B, Venugopal A, Balakrishnan L, Marimuthu A, Banerjee S, Somanathan DS, Sebastian A, Rani S, Ray S, Harrys Kishore CJ, Kanth S, Ahmed M, Kashyap MK, Mohmood R, Ramachandra YL, Krishna V, Rahiman BA, Mohan S, Ranganathan P, Ramabadran S, Chaerkady R, Pandey A (2009) Human Protein Reference Database – 2009 update. Nucleic Acids Res 37(Database Issue):767–772
144. Breitkreutz BJ, Stark C, Reguly T, Boucher L, Breitkreutz A, Livstone M, Oughtred R, Lackner DH, Bähler J, Wood V, Dolinski K, Tyers M (2008) The BioGRID Interaction Database: 2008 update. Nucleic Acids Res 36(Database Issue):637–640.
145. Blake JA, Dolan M, Drabkin H, Hill DP, Ni L, Sitnikov D, Burgess S, Buza T, Gresham C, McCarthy F, Pillai L, Wang H, Carbon S, Lewis SE, Mungall CJ, Gaudet P, Chisholm RL, Fey P, Kibbe WA, Basu S, Siegele DA, McIntosh BK, Renfro DP, Zweifel AE, Hu JC, Brown NH, Tweedie S, Alam-Faruque Y, Apweiler R, Auchinchloss A, Axelsen K, Argoud-Puy G, Bely B, Blatter M, Bougueleret L, Boutet E, Branconi S, Breuza L, Bridge A, Browne P, Chan WM, Coudert E, Cusin I, Dimmer E, Duek-Roggli P, Eberhardt R, Estreicher A, Famiglietti L, Ferro-Rojas S, Feuermann M, Gardner M, Gos A, Gruaz-Gumowski N, Hinz U, Hulo C, Huntley R, James J, Jimenez S, Jungo F, Keller G, Laiho K, Legge D, Lemercier P, Lieberherr D, Magrane M, Martin MJ, Masson P, Moinat M, O'Donovan C, Pedruzzi I, Pichler K, Poggioli D, Porras Millan P, Poux S, Rivoire C, Roechert B, Sawford T, Schneider M, Sehra H, Stanley E, Stutz A, Sundaram S, Tognolli M, Xenarios I, Foulger R, Lomax J, Roncaglia P, Camon E, Khodiyar VK, Lovering RC, Talmud PJ, Chibucos M, Gwinn Giglio M, Dolinski K, Heinicke S, Livstone MS, Stephan R, Harris MA, Oliver SG, Rutherford K, Wood V, Bahler J, Lock A, Kersey PJ, McDowall MD, Staines DM, Dwinell M, Shimoyama M, Laulederkind S, Hayman T, Wang S, Petri V, Lowry T, D'Eustachio P, Matthews L, Amundsen CD, Balakrishnan R, Binkley G, Cherry JM, Christie KR, Costanzo MC, Dwight SS, Engel SR, Fisk DG, Hirschman JE, Hitz BC, Hong EL, Karra K, Krieger CJ, Miyasato SR, Nash RS, Park J, Skrzypek MS, Weng S, Wong ED, Berardini TZ, Li D, Huala E, Slonim D, Wick H, Thomas P, Chan J, Kishore R, Sternberg P, Van Auken K, Howe D, Westerfield M (2012) The Gene Ontology: enhancements for 2011. Nucleic Acids Res 40(Database Issue):559–564
146. Zheng J, Zhang D, Przytycki PF, Zielinski R, Capala J, Przytycka TM (2009) SimBoolNet – a Cytoscape plugin for dynamic simulation of signaling networks. Bioinformatics 26:141–142
147. Erhard F, Friedel CC, Zimmer R (2008) FERN – a Java framework for stochastic simulation and evaluation of reaction networks. BMC Bioinform 9:356
148. Takahashi K, Kaizu K, Hu B, Tomita M (2004) A multi-algorithm, multi-timescale method for cell simulation. Bioinformatics 20(4):538–546
149. Janowski S, Kormeier B, Töpel T, Hippe K, Hofestädt R, Willassen N, Friesen R, Rubert S, Borck D, Haugen P, Chen M (2010) Modeling of cell-to-cell communication processes with Petri Nets using the example of Quorum sensing. In Silico Biol 10:27–48
150. Kormeier B, Hippe K, Arrigo P, Töpel T, Janowski S, Hofestädt R (2010) Reconstruction of biological networks based on life science data integration. J Integr Bioinform 7(2):146–159
151. Sommer B, Tiys ES, Kormeier B, Hippe K, Janowski SJ, Ivanisenko TV, Bragin AO, Arrigo P, Demenkov PS, Kochetov AV, Ivanisenko VA, Kolchanov NA, Hofestädt R (2010) Visualization and analysis of a cardio vascular disease- and MUPP1-related biological network combining text mining and data warehouse approaches. J Integr Bioinform 7:148
152. Proß S, Janowski SJ, Hofestädt R, Bachmann B (2012) A new object-oriented Petri net simulation Environment Based On Modelica. In: Online proceedings of the 2012 Winter simulation conference, Berlin. IEEE, pp 1–13
153. Klenke C, Janowski S, Borck D, Widera D, Ebmeyer LA, Kalinowski J, Leichtle A, Hofestädt R, Upile T, Kaltschmidt C, Kaltschmidt B, Sudhoff H (2012) Identification of novel cholesteatoma-related gene expression signatures using full-genome microarrays. PloS One 7(12):1–14
154. Association M (2005) Modelica – a unified object-oriented language for physical systems modeling. Lang Specif 2:7–11