match the different components of the design phase. Further, an assembler would allow the assembly of the different components and allow further simulation iterations before final fabrication.

As seen from the workflow, modelling and simulation, together with standardized visual representation and the ability to store and share the biological circuit in a machine-readable format, are key elements of the process. Several challenges exist in integrating standardized model visualization and simulation techniques in the workflow.

First, visualization standards should have the capability to capture biological entities at different granularities, from promoter sequences, ribosome-binding site sequences, genes, proteins to higher-order molecular pathways and complexes.

Second, simulation tools should be able to define models at these multiple scales, capturing unknown interactions and parameters while integrating standardized, characterized biological components into networks of interacting modules.

Third, there exists a knowledge gap between visual representations of biological systems vis-à-vis corresponding mathematical models. While graphical representations (molecular maps) are intended to capture the known biology of processes, mathematical models encode only parts of the detailed molecular maps owing to the underlying complexity of the interactions, insufficient data to characterize processes, unknown parameters or shortcomings in mathematical representation of complex interactions. *In silico* modelling and simulation tools should provide mechanisms to reconcile such knowledge gaps in their framework. Moreover, software tools should provide mechanisms for incorporating restrictions in parts assembly and design phases of biological circuits.

In this respect, development of a common framework, encompassing different tools and schemata employed in the different phases, is required to accelerate the progress of synthetic biology. To share the results of modelling efficiently, we need a common language in representing: (i) mathematical contents; (ii) semantics, annotations; and (iii) visual representation of models.

Figure 1 shows a schematic of the biological engineering process with different schemata available for standardized exchange of information. As seen from the figure, standards encompassing mathematical content, like the Systems Biology Markup Language (SBML; Hucka *et al.* 2003), semantics and annotation, like Minimum Information Requested in the Annotation of biochemical Models (MIRIAM; Le Novère *et al.* 2005), and visualization, like the Systems Biology Graphical Notation (SBGN; Le Novère *et al.* 2008), can be employed for modelling and simulation of biological parts, devices as well as systems. In order to facilitate *in silico* simulation, a common language for the simulation systems would facilitate the sharing of results and usage of different simulation engines. Efforts are now being made to standardize simulation result description such as SBRML (Dada *et al.* 2009), which can be integrated into the workflow as depicted in figure 1. The figure illustrates how modelling and simulation techniques can be employed at each step of the design and assembly process hierarchy of biological parts, devices and systems. However, such standards need to be enhanced to address the unique challenges of synthetic biology elucidated earlier.

In the remaining sections, we outline the different standardization efforts in the systems and synthetic biology communities and provide potential avenues for developing a common framework in a mutually beneficial manner.

3. STANDARDS AND TOOLS IN BIOLOGICAL ENGINEERING

3.1. Efforts in synthetic biology

The goal in synthetic biology is to develop engineered biological circuits with well-defined input–output behaviour. Thus, design, simulation and assembly tools that aid in the development of high-precision biological constructs form an integral part of this effort. Moreover, the ability to *re-use* well-characterized parts is a hallmark of any engineering discipline. In this respect, significant efforts have been undertaken in the standardization of biological parts and their systematic storage and retrieval, which we briefly review next.

As synthetic biology is an engineering discipline from the onset, efforts in standardization were in place at the very early stages of research activities. The Registry of Standard Biological Parts (http://parts.mit.edu/) is a database to create and maintain such building blocks, providing free access to an 'open-commons' of basic biological functions. The goal is to streamline the fabrication of complex constructs to programme synthetic biological systems. The registry stores the biological constructs, providing detailed *worksheets* of the input–output characteristics of biological components together with the *interfaces* for communication between them. It contains *biological parts*, which are combined to form *biological devices* that can be further connected to build engineered *biological systems*.

While the registry provides storage of the biological components, their properties and interfaces need to be defined in terms of standardized schemata that allows unambiguous definition of the parts and their behaviour. In this effort, the BioBrick parts, an open source genetic parts as defined via an open technical standards setting process that is led by the BioBricks Foundation (http://bbf.openware.org), represent an effort to introduce the engineering principles of abstraction and standardization into synthetic biology. BioBricks standard biological parts are DNA sequences of defined structure and function; they share a common interface and are designed to be composed and incorporated into living cells such as *Escherichia coli* to construct new biological systems.

3.1.1. Challenges in parts characterization and assembly. One of the major challenges faced by the synthetic biology community with regard to standard formation is how to characterize parts features. While it is

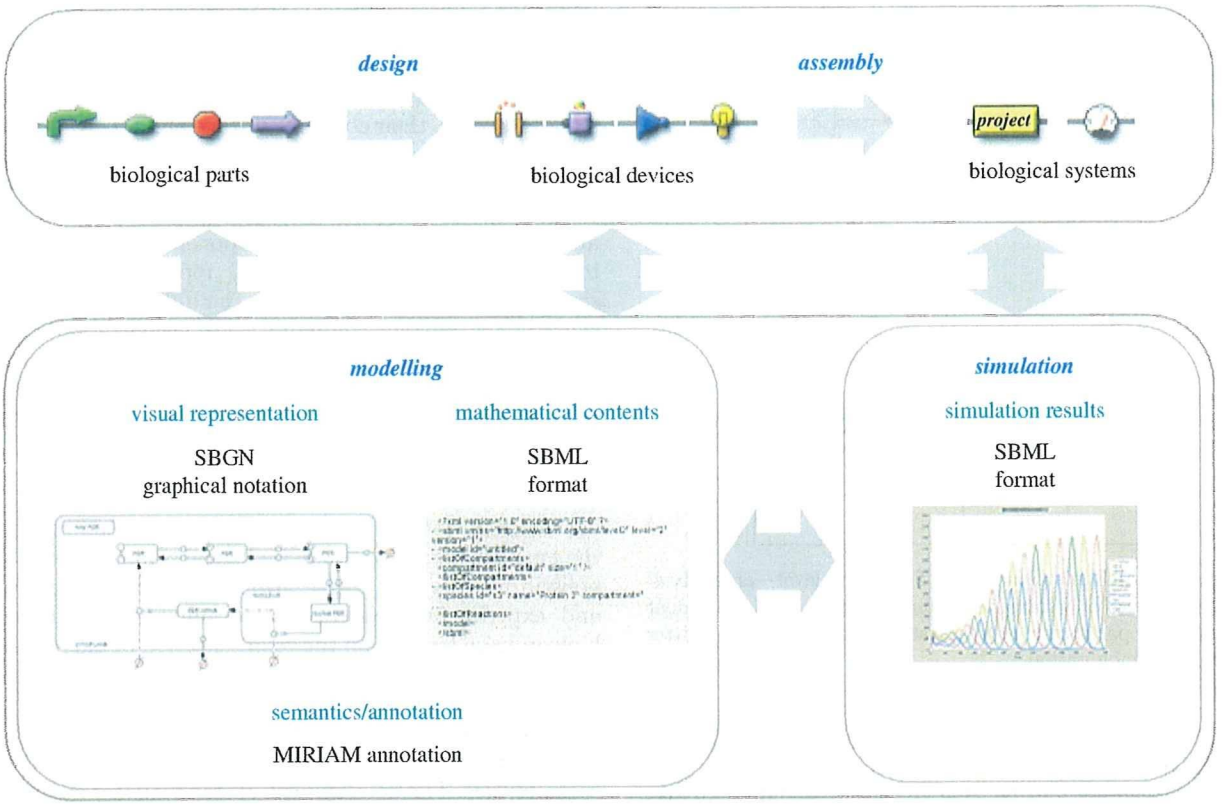

Figure 1. Workflow for the design and development of engineered biological circuits.

currently defined in terms of promoter structure and sequences, it is not a characterization in terms of function in the context of interacting networks.

A proper level of description that may smoothly interface with a network-level context is at the device behaviour level. This is at the same level of description as in electric design. In electric design, apart from circuit diagrams, there are datasheets for each component that specify basic parameters and modes of action. For example, in transistor specification, datasheets define various basic parameters and behaviour characteristics of a transistor. Such specifications can be used to generate properly parametrized equivalent circuits that provide a versatile definition of functional behaviour in a specific network context. Figure 2*a* is an illustrative example of an equivalent circuit at the device level, and figure 2*b* is an equivalent circuit of common emitter configuration of the same device in a network context. In the circuit-level equivalent circuit, parameters such as h_{fe}, i.e. current amplification ratio, and h_{ie}, i.e. input impedance, are defined. Electronic engineers can design and analyse circuits using these parameters without looking into details of implementations. This representation significantly enhanced our capability to design and analyse circuits and is particularly important when it has to be scaled up. At the same time, it should be noted that unlike electronic circuits in which an identical device can be used in multiple places in the system (such as using FET-type 2SK30A in different amplification modules), a synthetic biology device can be used only once in the system to avoid unexpected interferences. Thus, 'device' in the synthetic biology context shall be considered as 'family of parts'. Therefore, it is essential for each device description to have parametric features reflecting different parts of the family.

The ability to define biological parts unambiguously at the device as well as network levels is one of the key challenges in synthetic biology. Current efforts to define interfaces to biological devices in a molecule-independent manner use parameters such as polymerase per second, which is the flow rate of RNA polymerase molecules along the DNA, or by ribosomes per second which measures the flow rate of ribosomes along the mRNA molecule. Figure 3*a* shows an illustrative device description of a banana odour generator from the BioBricks registry. In this figure, ATF1 transcription activity is described as a function of input signal that activates ATF1 transcription. The output, isoamyl acetate production, is described as a function of isoamyl alcohol and ATF1 transcription activity. In order to characterize the device, it is required to define a parameter capturing the function of the device, instead of the kinetic constant of ATF1 catalysing isoamyl alcohol conversion into isoamyl acetate. It is not always useful to describe the kinetic constant for ATF1 enzyme alone because BioBrick is assumed to be the building blocks at the device level. Thus, biological equivalents of parameters such as h_{fe} need to be defined that provide a rate at which changes in signalling to transcription of ATF1 affect the rate of isoamyl acetate production (e.g. k_{sp}: signal–product amplification rate). Also, it should have a parameter (k_{ip}: input–product rate) that characterizes change in product (isoamyl acetate) per change in input (isoamyl alcohol). The above example is only provided to describe the level

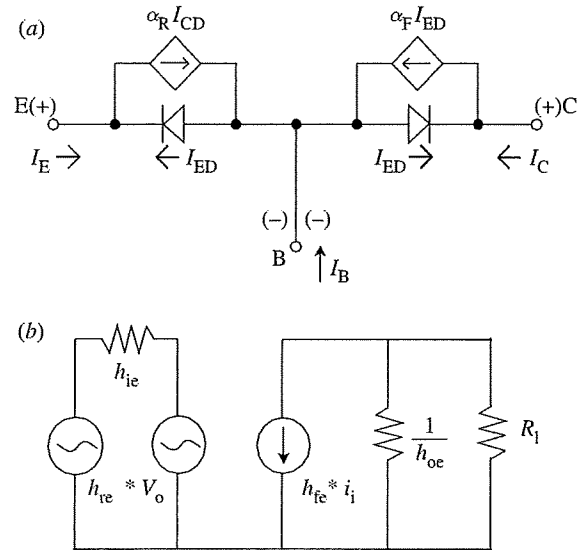

Figure 2. (*a*) Equivalent circuit for device level—an equivalent circuit for NPN transistor (from Wikipedia). (*b*) Equivalent circuit (*h* parameter representation) for common emitter topology configuration.

of abstraction at which an equivalent biological circuit shall be defined (figure 3*b*). With these parameters, BioBrick designers will be able to design scalable circuits without examining details of biological elementary reactions. This example highlights the need for developing a standard format for describing elementary building blocks at the device level, which can be integrated in synthetic biology parts databases, like BioBricks.

3.1.2. Software support in synthetic biology. With the development of parts databases on the one hand, efforts to build software support for synthetic biology are also underway. Software platforms such as BioJADE (Goler 2004), Atiiena, now called TinkerCell (Chandran *et al.* 2009), and GenoCAD (Cai *et al.* 2007) have been designed specifically for synthetic biology needs. Tools like GeneDesign (Richardson *et al.* 2006) provide a genome version control system, while BrickIt and Clotiio provide mechanisms to manipulate DNA and protein coding sequences. Machine-readable language efforts have also been developed in Antimony and LBS. Table 1 provides an overview of the common software tools and platforms in synthetic biology.

These tools focus on specific aspects of sequence design with minimal information on interactions at a network level. These will be powerful and useful tools for designing systems on a small scale, where possible interactions can be intuitively followed. However, scaling up of synthetic biology modules to form network elements will quickly make dynamical behaviours intractable. Moreover, higher level descriptions need to be developed at the level of device function and mode of action as device-level part characterizations become more commonplace. It is envisaged that as building blocks are assembled and scaled up to relatively complex networks, computer-assisted network design and modelling tools such as CellDesigner (Funahashi *et al.* 2003) would be useful to capture network dynamics generated from assembled parts.

The efforts outlined earlier focus on the definition of standard parts, their common storage and retrieval and tools to simulate them. However, a significant aspect of biological engineering is the visual representation of the components (from parts to devices and finally systems), ability to simulate different components *in silico* and tools to support such graphical representation and simulation. In §3.2, we overview the different efforts in the systems biology community for visualization and simulation standards before providing insights into the development of a common computational framework across the synthetic and systems biology domains.

3.2. Efforts in systems biology

The focus in systems biology has been on understanding the mechanistic behaviour of biological components in a holistic manner and aggregating data from literature and experimental systems. Major efforts in systems biology have focused on the representation of biological pathways and their molecular interactions, together with the development of simulation schemata. Pathway standards have been developed, aiming to facilitate collaboration and data exchange among various research communities. The development of standards for computational platforms in biological engineering can be broadly defined in terms of four key feature elements, which we elucidate next:

(i) standardization of representation of mathematical contents,
(ii) semantics and annotations,
(iii) visual representation of models, and
(iv) simulation of biochemical networks.

3.2.1. Standardization of representation of mathematical contents. With the rapid increase in the volume of high throughput data available to systems biologists, efforts for defining standards for storage, analysis and exchange of large datasets have been undertaken on a community-wide scale. Standards include Gene Ontology (Ashburner *et al.* 2000) for describing gene functions, SBML and CellML (Lloyd *et al.* 2008) for describing biochemical reaction networks and Minimum Information About a Simulation Experiment (MIASE, http://www.ebi.ac.uk/compneur-srv/miase/), to name a few. While the goal of all these standards (extensively reviewed in Brazma *et al.* 2006; Strömbäck *et al.* 2007) is to define a consistent schema for data exchange, individual standards are targeted towards addressing specific issues. The ability to represent biological knowledge in a mathematically consistent format is key to performing *in silico* simulation and analysis of their dynamic behaviours. In this respect, SBML and CellML are two standards adopted across the systems biology community.

SBML is a machine-readable format for representing pathway models (http://sbml.org; Hucka *et al.* 2003).

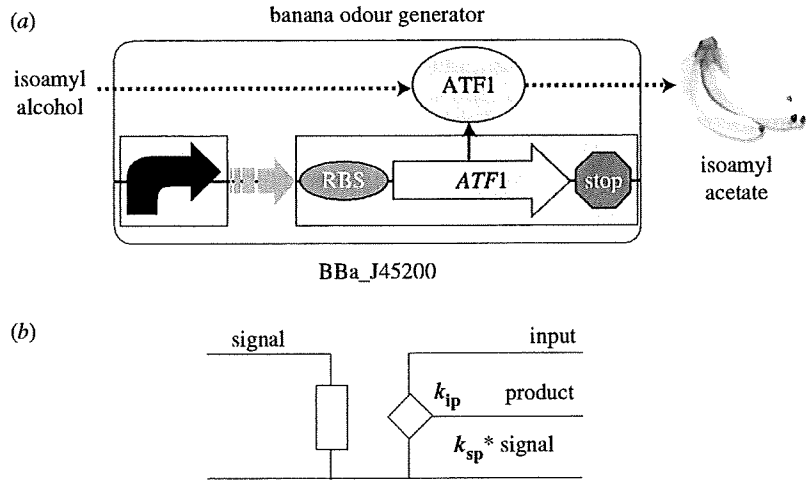

Figure 3. (*a*) An example of device description from BioBrick (taken from BioBrick registry). (*b*) An example of equivalent circuit for BBa_J45200. Shown only to provide the sense of abstraction, instead of exact formalism.

Table 1. Software tools and standards in synthetic biology.

software tool	overview	references	availability
ASMPARTS	a computational tool to produce models of biological systems by assembling models from biological parts	Rodrigo *et al.* (2007*a*,*b*)	http://soft.synth-bio.org/asmparts.html
Antimony	a human-readable and human-writable language for describing biological modules		http://antimony.sourceforge.net
BIOJADE	a design and simulation tool for synthetic biological systems	Goler (2004)	http://web.mit.edu/jagoler/www/biojade
BIOMORTAR	a laboratory management system designed specifically to deal with BioBricks		http://igem.uwaterloo.ca/BioMortar
BRICKIT	portable Web-based registry that helps synthetic biologists to plan, organize and track their local BioBrick samples		http://brickit.wiki.sourceforge.net
CLOTHO	a design environment to manipulate DNA sequence information and store the manipulated data as packaged 'parts' back to part repositories		http://biocad-server.eecs.berkeley.edu/wiki/index.php/Clotho_Development
GENEDESIGN/ BIOSTUDIO	a suite of algorithms that allow users to edit several features of protein coding sequences in an integrated development environment with a genome version control system	Richardson *et al.* (2006)	http://www.genedesign.org
GENETDES	a tool to design transcriptional networks with targeted behaviour	Rodrigo *et al.* (2007*a*,*b*)	http://soft.synth-bio.org/genetdes.html
GENOCAD	a Web-based application guiding users through the design of part-based genetic systems	Cai *et al.* (2007)	http://www.genocad.org
LBS	a language for biological systems. It aims to provide a language for programming systems at the logical level of interactions between genes and proteins		http://www.inf.ed.ac.uk/publications/report/1270.html
OPENCELL (PCENV)	an environment for creating and simulating arbitrary mathematical models	Beard *et al.* (2009)	http://www.cellml.org/tools/opencell
SYNBIOSS	a software suite for the quantitative simulation of biochemical networks using hybrid stochastic algorithms	Hill *et al.* (2008)	http://synbioss.sourceforge.net
TINKERCELL (ATHENA)	a tool for building, simulating and analysing genetic circuits	Chandran *et al.* (2009)	http://www.tinkercell.com

It was developed by an international community of systems biologists and software developers aiming to provide a common intermediate format for data sharing among various computer modelling software applications. SBML is neutral with respect to programming languages and software encoding; however, it is encoded using XML (Bray *et al.* 2008). By supporting SBML as a format for reading and writing models,

different software tools (including programs for building and editing models, simulation programs, databases and other systems) can directly communicate and store the *same* computable representation of those models. Currently, there are over 160 software packages supporting SBML (http://sbml.org/SBML_Software_Guide/SBML_Software_Summary).

Another major standardization effort for machine-readable representation of biological pathways is CellML (http://www.cellml.org; Lloyd *et al.* 2008). It is an XML-based markup language originally developed by the Auckland Bioengineering Institute at the University of Auckland and affiliated research groups. It is similar to SBML, but more suited for multi-scale biological modelling capturing the structure and underlying mathematics of cellular models in a generic manner. CellML is growing in popularity as a portable description format for computational models, and groups throughout the world are using CellML for modelling or developing software tools based on it. Currently, there are a set of open-access tools and model storage databases based on CellML available at http://www.cellml.org/tools, including VIRTUAL CELL (Loew & Schaff 2001), a Java-based modelling and simulation environment that imports and exports CellML.

While the current standards are largely geared towards mathematical modelling of biological pathways and molecular interactions, it is envisaged that future developments would be able to incorporate synthetic biology constructs, such as protein coding sequences, biological parts and device-level information in their formalism. Some software tools, such as OPENCELL/PCENV (table 1), support mathematical modelling for systems and synthetic biology constructs while using CellML as the native format for model storage. Similar efforts to enhance existing standards and synergize their usage would accelerate the adoption of consistent standards in biological engineering.

3.2.2. Semantics and annotation. For models to be informative, they need to be properly annotated with sufficient information attached to them enabling third parties to effectively use such models. BioPAX (http://www.biopax.org) is a collaborative effort to create a data exchange format for biological pathway data with ontological annotations. The main purpose is to facilitate data access, sharing and integration from multiple pathway databases by biologists. BioPAX supports representation of metabolic and signalling pathways, molecular and genetic interactions and gene regulation. Relationships between genes, small molecules, complexes and their states (e.g. post-translational protein modifications, mRNA splice variants, cellular location) are described, including biological events. BioPAX is complementary to other standard pathway information exchange languages, including SBML and CellML, as it focuses on large qualitative pathways and their integration rather than on mathematical modelling.

Since BioPAX targets annotation on pathways for existing organisms, it does not directly translate into the synthetic biology domain. However, BioPAX-like ontological annotation may need to be initiated for BioBricks. This is important as the semantics of each device (BioBricks) may become unmanageable once engineered circuits exceed certain levels of complexity and millions of variations arise for similar functional devices. The current registry of biological parts can be a potential starting point for such ontological annotations. Other effort in annotation, particularly for model annotations, exists in the MIRIAM project (http://www.ebi.ac.uk/miriam/; Le Novère *et al.* 2005). The MIRIAM standard defines the minimum information that has to be attached to the model so that the model can be informative by itself.

The Systems Biology Ontology (SBO; http://www.ebi.ac.uk/sbo/main/) project endeavours to enhance the semantics of models, regardless of modelling approaches (refer to Brazma *et al.* (2006) and Strömbäck *et al.* (2007), for details on semantics approaches in systems biology). Again, standards like MIRIAM and SBO, although designed for molecular network models, can be extended to synthetic biology devices and circuit descriptions, particularly to define standard semantics for biological parts and devices.

3.2.3. Visual representation of models. Clear and unambiguous visualization is a fundamental step in applying computational techniques to biological models for scientific discovery. It is important to have standard visual representation languages for describing events and concepts in biology, such as biochemical interaction network, inter- and intracellular signalling and gene regulation. Most of the field is permeated with ad hoc graphical notations that have little in common between different researchers, publications, textbooks and software tools. While simplified notations can be used for purposes of elucidation, standardized representation of biological entities is of paramount importance for exchange between computational tools.

Thus, it is imperative to define a comprehensive set of graphical symbols that have precise semantics and detailed syntactic rules defining their use and are insensitive to restrictions of any medium or software, so that they can be used across a large array of applications to enhance data sharing, exchange and integration.

Definition of a common lingua franca is an important step in the standardization of biological representations and technologies. Biology has traditionally been a descriptive science, where the role of pictures and diagrams cannot be overstated. A community-wide effort is currently underway to define the SBGN (http://sbgn.org). The goal of SBGN is to define a set of visual glyphs and syntax, so that anyone can understand what the diagram exactly means much in the same vein as electrical circuit diagrams used by chip designers. The SBGN project was initiated by a group of biochemists, molecular biologists, modellers and computer scientists, with the aim of developing and standardizing a systematic and unambiguous graphical notation for applications in systems biology. SBGN is expected to be used not only by systems biologists,

but also by biologists of all disciplines, educators, publishers and students. Level 1 specification of the SBGN process diagram was released in August 2008 (Le Novère *et al.* 2008). The SBGN entity relationship diagram and activity flow diagram specifications now exist as draft proposals and are expected to be released in 2009. Currently, SBGN specification only defines visual icons and their syntax. Specification of the file format on how such graphics shall be stored and exchanged is the subject of future development.

Another effort in the direction of visualization is the SBML layout extension (Gauges *et al.* 2006). While the SBML file format does not provide for the storage of visual information for reaction graphs, the extension aims to provide a schema for describing the position and size of objects associated with biochemical reactions, thus providing the potential to render complex graphical standards with the SBML schema.

Early efforts to develop consistent schematics for synthetic biology constructs are represented by BOGL (http://openwetware.org/wiki/Endy:Notebook/BioBrick_Open_Graphical_Language), a graphical language for the formal description of standard biological parts. It aims to define symbolic notations for different biological parts, such as selection markers and restriction markers (Shetty *et al.* 2008). The standardization of visual representations in biological engineering presents new areas of research, particularly in enhancing existing standards to incorporate multi-level views—from detailed sequence level, binding site domain level views to molecular interactions, pathways and large-scale networks, in a consistent format. Visualization tools in the biological domain need to support such a *semantically zoomable* (Hu *et al.* 2007) multi-dimensional view of biological parts, devices and systems in the future.

3.2.4. Simulation of biochemical networks. While standardization is an integral part of the process of computational systems biology, the development of software tools for model building, distribution and running simulations is another important dimension. In this direction, plenty of model building and simulation tools are available. CELLERATOR (http://www.cellerator.org; Shapiro *et al.* 2003), COPASI (http://www.copasi.org; Hoops *et al.* 2006) and DIZZY (http://www.systemsbiology.org/Technology/Data_Visualization_and_Analysis/Dizzy; Ramsey *et al.* 2005) in the academic community and SIMBIOLOGY from Mathworks Inc. (http://www.mathworks.com/products/simbiology/) and PHYSIOLAB (Entelos Inc.; http://www.entelos.com/physiolabModeler.php) exist, each catering to different modelling techniques (refer to Hucka *et al.* (2004) for a review on SBML compliant simulators).

One of the most popular and widely used tools in this category is CELLDESIGNER (Funahashi *et al.* 2003)—a modelling and simulation tool to visualize, model and simulate gene-regulatory and biochemical networks. Two major characteristics embedded in CELLDESIGNER boost its usability to create/import/export models: (i) solidly defined and comprehensive graphical representation, specifically process diagram-based notations (Kitano *et al.* 2003; Kitano *et al.* 2005) of network models and (ii) SBML as a model-describing basis, which functions as inter-tool media to import/export SBML-based models. Moreover, CELLDESIGNER provides the ability to embed or smoothly connect via Systems Biology Workbench (Sauro *et al.* 2003) different simulation/analysis packages that allow the simulation of the pathways using various simulation techniques such as COPASI, SBML ODE SOLVER (Machné *et al.* 2006), etc.

The simulation tools in the systems biology space currently focus on simulation of biological pathways and networks represented as biochemical reactions. On the other hand, simulation tools in synthetic biology allow the study of the dynamic behaviour of specific building blocks (like transcription constructs). As mentioned earlier, large-scale assembly of biological constructs would require multi-level modelling and simulation capabilities. For example, CELLDESIGNER currently does not have the capability to assist parts design and simulation. However, the software supports a plug-in architecture through which it is possible to establish close links with tools such as BIOJADE or to develop plug-ins for assembly and design of biological parts. Such synergistic integration across tools is essential to create consistent design platforms for synthetic biology.

4. BUILDING A COMMON FRAMEWORK

As reviewed in the previous section, various standards and technologies are already in place for practical use to cope with distinct levels of biological modelling and simulation. The goal of standardization is to fit together different pieces in a consistent manner to build a useful whole (Brazma *et al.* 2006). Integration of the various approaches and efforts to build a common framework to share accumulating knowledge in models is critical for advance in biological science across various disciplines. However, as elucidated in previous sections, several challenges need to be addressed to enhance existing standards in mathematical representations, visualization and modelling tools to accommodate the unique features of synthetic biology.

For representation and exchange of biochemical network models, SBML can be used as a good medium. The essential strength of the SBML format lies in the simulation of biological networks. SBML is defined as a set of standards to facilitate effective and efficient sharing of models with biochemical reactions. As mentioned earlier, semantic annotation of biological models is an important step in developing consistent, unambiguous representation. SBML, in conjunction with annotation standards such as MIRIAM, provides the ability to store and share information in a seamless, unambiguous fashion, which can be used as a medium for study and analysis in both the synthetic and systems biology communities.

There have already been some attempts to provide the tools for synthetic biological modelling, which allow converting the models into SBML format, such as ATHENA (Chandran *et al.* 2009) and ASMPARTS (Rodrigo *et al.* 2007*a*,*b*).

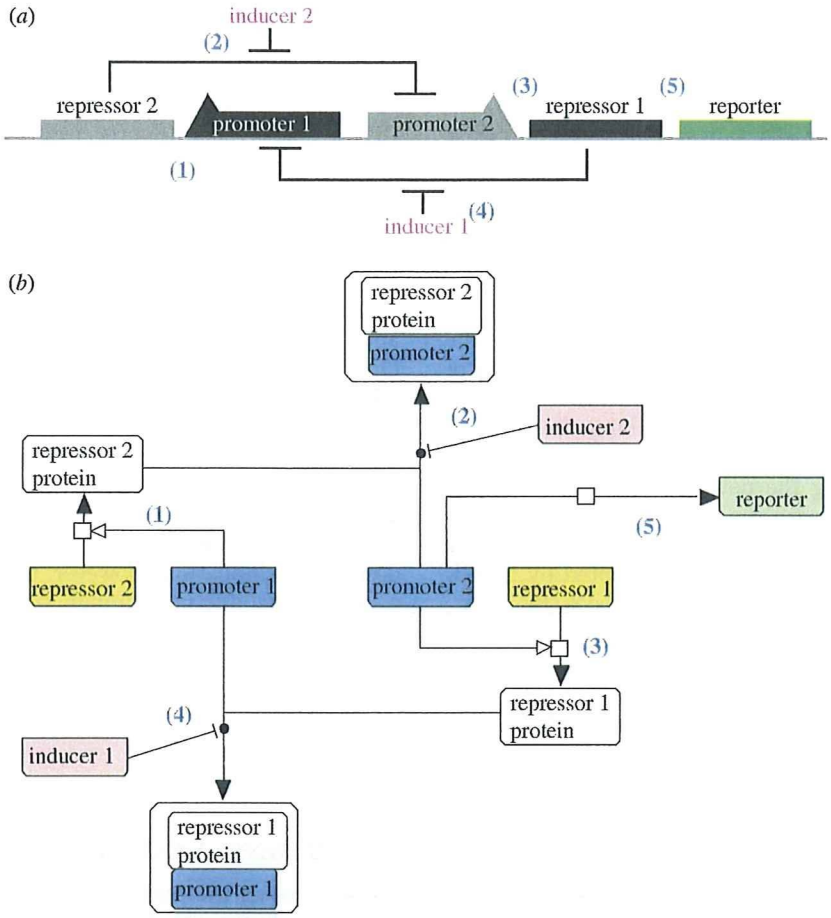

Figure 4. (*a*) Genetic toggle switch design taken from Gardner *et al.* (2000). (*b*) A possible representation of the genetic toggle switch in the process diagram format. Numbers in parentheses show the corresponding 'processes' in both (*a*) and (*b*) diagrams. (1) Promoter 1 promotes repressor 2 translation, (2) inducer 2 induces the promoter 2 activities, (3) promoter 2 promotes repressor 1 translation, (4) inducer 1 induces the promoter 1 activities, (5) promoter 2 promotes reporter translation. Arrows with filled heads describe the state transition, while arrows with open heads describe the stimulation to the process. T-shaped arcs represent inhibition.

There are several models that have been converted into SBML format and curated and stored in public databases. Elowitz & Leibler's (2000) classic 'repressilator' model, for example, is already registered and available at BioModels database (http://biomodels.net; Le Novère *et al.* 2006). While efforts are underway to convert models of biological circuits into SBML format (Rodrigo *et al.* 2007*a*,*b*; Chandran *et al.* 2009), which can then be simulated using various SBML compliant simulation tools, a concerted effort is imperative to provide consistent visual representation of the various BioBricks—biological parts, devices and systems.

On the other hand, on-going efforts in the SBGN community provide a platform for defining a common visual language for biological systems—from engineered circuits to cellular pathways. The SBGN schema envisages supporting the representation of systems at different scales through process diagrams, activity flow or entity relationship diagrams. These different diagrammatic schemata allow the representation of knowledge at different levels of abstraction depending on the scope, accuracy of knowledge and other design requirements.

We provide some illustrative examples of possible representation of classical biological constructs in current process diagram notation. In the process diagram, nodes represent the states of biological entities and arcs describe biological process between the states. As a first example of developing an SBGN process diagram compliant representation of synthetic biological circuits, we consider the classical toggle switch model by Gardner *et al.* (2000). The genetic toggle switch model, constructed by Gartner *et al.*, toggles between stable transcription from either of two promoters in response to external signals (figure 4*a*,*b*). While figure 4*a* captures the toggle switch behaviour of the system, figure 4*b* endeavours to give a more mechanistic view, showing, for example, the mechanism of repression of the promoters by complex formation with the repressors. It is possible to capture different mechanisms of repression in the process diagram notation, where the biology of the process is known.

Another example illustrated here is a translational switch (Isaacs *et al.* 2004). While the classical diagram (figure 5*a*) captures the structural changes of the entities in the biological process, the process diagram

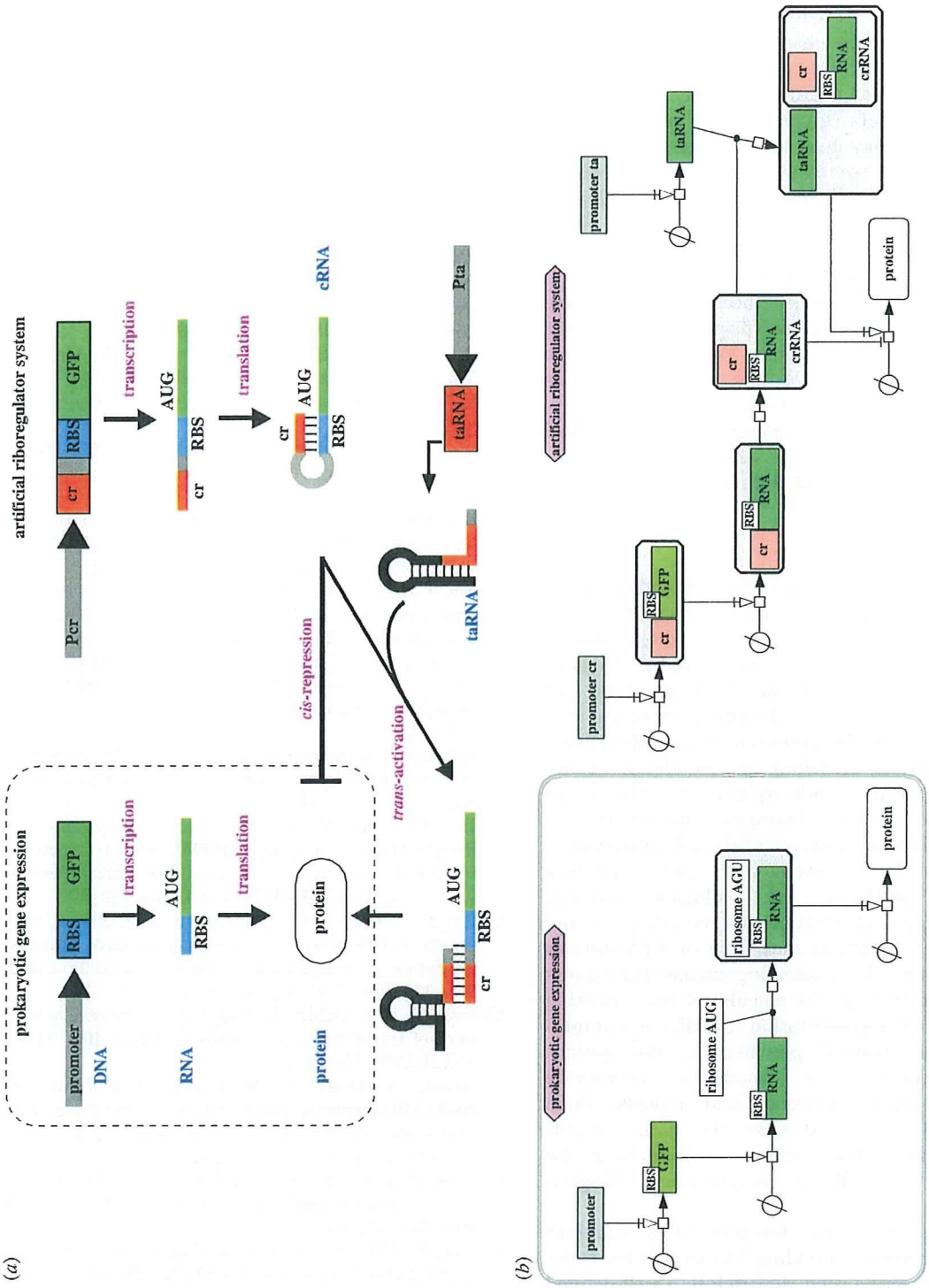

Figure 5. (*a*) An example of translational switch (Isaacs *et al.* 2004). (*b*) An example of equivalent riboregulator system represented in the process diagram format. (*a*) The artificial riboregulator system, shown in the graphical representation typically adopted by synthetic biology, and (*b*) interpretation in the process diagram notation. The circle with slash glyph represents the source of the creation of the entity or a state. The arrow with bar indicates necessary stimulation to the process. Lines connected with black dots represent the binding (e.g. RNA and ribosome, taRNA and crRNA).

(figure 5*b*) clearly identifies each step of the event and illustrates the mechanism of causes and effects of the processes.

As can be observed from these diagrams (figures 4 and 5), the genetic toggle switch as well as translational switch can be represented in process diagram manner. As both diagrams (figures 4*b* and 5*b*) are constructed in CellDesigner and stored in the SBML format, it is possible to simulate the characteristics of the models once the dynamics of the processes is described in mathematical formulae in SBML format.

The visual elements (glyphs) in the current SBGN standard proposal may need to be enhanced to accommodate the different components used in the synthetic biology community, as the current standard evolves in an inter-community-wide collaborative manner. Sharing of symbols representing identical biological elements would further help in developing a common graphical lingua franca for biological engineering, on the same lines as in electrical circuit diagrams and other advanced engineering disciplines. We strongly believe that careful collaboration on the visual as well as model representation aspects between the two communities would foster the development of a standard graphical notation schema and accelerate the application of computational techniques.

5. SUMMARY

While the paradigm of systems biology endeavours a *holistic* understanding of the working principles of complex biological networks, *discovery by design* forms a key essence in synthetic biology, motivated by Richard Feynman's phrase 'What I cannot create, I do not understand' (Simpson 2006). In this perspective, the two disciplines hold the potential of complementing each other—analysis, modelling and simulation of biological networks can provide insights into the design and synthesis of de novo biological circuits. In this article, we provided an overview of the role of standardization in developing systematic and consistent computational platforms for biological systems. Particularly, graphical notations for visualization and schemata for mathematical modelling of such systems will play a pivotal role in enforcing engineering rigours in the study of biology. As elucidated here, existing standards of model representation (SBML) and graphical visualization (SBGN) prevalent in the systems biology community can be extended to incorporate synthetic biological constructs and models. Such collaborative efforts would pave the path towards a common, standardized schematic framework for understanding as well as engineering biological systems.

At the same time, development of a standard specification for genetic building blocks will force the community to describe each BioBrick in a well-defined form as exemplified in the equivalent circuit concept in electronics. Such a practice will not only benefit the synthetic biology community, but also the systems biology community because it triggers accumulation of knowledge on canonically defined genetic circuits. When synthetic biology matures as an engineering field, the issues discussed in this paper will be the common practice and that is when it can be regarded as precision engineering.

This research is, in part, supported by ERATO-SORST Program of the Japan Science and Technology Agency (JST), Genome Network Project of the Ministry of Education, Culture, Sports, Science and Technology, NEDO Fund for International Standard Formation from the New Energy Development Organization and the Okinawa Institute of Science and Technology.

REFERENCES

Ashburner, M. *et al.* 2000 Gene Ontology: tool for the unification of biology. *Nat. Genet.* **25**, 25–29. (doi:10.1038/75556)

Beard, D. A. *et al.* 2009 CellML metadata standards, associated tools and repositories. *Phil. Trans. R. Soc. A* **367**, 1845–1867. (doi:10.1098/rsta.2008.0310)

Bray, T., Paoli, J., Sperberg-McQueen, C. M., Maler, E. & Yergeau, F. (eds) 2008 Extensible Markup Language (XML) 1.0, 5th edn. See http://www.w3.org/TR/REC-xml/.

Brazma, A., Krestyaninova, M. & Sarkans, U. 2006 Standards for systems biology. *Nat. Rev. Genet.* **7**, 593–605. (doi:10.1038/nrg1922)

British Telecommunications 2007 Pharma futurology: joined-up healthcare, 2016 and beyond. See http://www2.bt.com/static/i/media/pdf/BT_Pharma_Lowres.pdf.

Brown, J. 2007 The iGEM competition: building with biology. *Synth. Biol.* **1**, 3–6. (doi:10.1049/iet-stb:20079020)

Cai, Y., Hartnett, B., Gustafsson, C. & Peccoud, J. 2007 A syntactic model to design and verify synthetic genetic constructs derived from standard biological parts. *Bioinformatics* **23**, 2760–2767. (doi:10.1093/bioinformatics/btm446)

Chandran, D., Bergmann, F. T. & Sauro, H. M. 2009 Athena: modular CAD/CAM software for synthetic biology. (http://arxiv.org/0902.2598)

Dada, J. O., Paton, N. W. & Mendes, P. 2009 Systems Biology Results Markup Language (SBRML) level 1: structure and facilities for results representation. See http://www.comp-sys-bio.org/static/SBRML-specs-15-04-2009.pdf

Deans, T. L., Cantor, C. R. & Collins, J. J. 2007 A tunable genetic switch based on RNAi and repressor proteins for regulating gene expression in mammalian cells. *Cell* **130**, 363–372. (doi:10.1016/j.cell.2007.05.045)

Elowitz, M. B. & Leibler, S. 2000 A synthetic oscillatory network of transcriptional regulators. *Nature* **403**, 335–338. (doi:10.1038/35002125)

Funahashi, A., Morohashi, M., Kitano, H. & Tanimura, N. 2003 CellDesigner: a process diagram editor for gene-regulatory and biochemical networks. *Biosilico* **1**, 159–162. (doi:10.1016/S1478-5382(03)02370-9).

Gardner, T. S., Cantor, C. R. & Collins, J. J. 2000 Construction of a genetic toggle switch in *Escherichia coli*. *Nature* **403**, 339–342. (doi:10.1038/35002131)

Gauges, R., Rost, U., Sahle, S. & Wegner, K. 2006 A model diagram layout extension for SBML. *Bioinformatics* **22**, 1879–1885. (doi:10.1093/bioinformatics/btl195)

Goler, J. A. 2004 BioJADE: a design and simulation tool for synthetic biological systems. Master's thesis, MIT Computer Science and Artificial Intelligence Laboratory, MIT-CSAIL-TR-2004-036.

Guido, N. J., Wang, X., Adalsteinsson, D., McMillen, D., Hasty, J., Cantor, C. R., Elston, T. C. & Collins, J. J.

2006 A bottom-up approach to gene regulation. *Nature* **439**, 856–860. (doi:10.1038/nature04473)

Hasty, J., McMillen, D. & Collins, J. J. 2002 Engineered gene circuits. *Nature* **420**, 224–230. (doi:10.1038/nature01257)

Hill, A. D., Tomshine, J. R., Weeding, E. M., Sotiropoulos, V. & Kaznessis, Y. N. 2008 SynBioSS: the synthetic biology modeling suite. *Bioinformatics* **24**, 2551–2553. (doi:10.1093/bioinformatics/btn468)

Hoops, S. *et al.* 2006 COPASI—a COmplex PAthway SImulator. *Bioinformatics* **22**, 3067–3074. (doi:10.1093/bioinformatics/btl485)

Hu, Z., Mellor, J., Wu, J., Kanehisa, M., Stuart, J. M. & DeLisi, C. 2007 Towards zoomable multidimensional maps of the cell. *Nat. Biotechnol.* **25**, 547–554. (doi:10.1038/nbt1304)

Hucka, M., Finney, A., Sauro, H., Bolouri, H. & Doyle, J. 2003 The Systems Biology Markup Language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics* **19**, 524–531. (doi:10.1093/bioinformatics/btg015)

Hucka, M. *et al.* 2004 Evolving a lingua franca and associated software infrastructure for computational systems biology: the Systems Biology Markup Language (SBML) project. *IEE Proc. Syst. Biol.* **1**, 41–53. (doi:10.1049/sb:20045008)

Isaacs, F. J., Hasty, J., Cantor, C. R. & Collins, J. J. 2003 Prediction and measurement of an autoregulatory genetic module. *Proc. Natl Acad. Sci. USA* **100**, 7714–7719. (doi:10.1073/pnas.1332628100)

Isaacs, F. J., Dwyer, D. J., Ding, C., Pervouchine, D. D., Cantor, C. R. & Collins, J. J. 2004 Engineered riboregulators enable post-transcriptional control of gene expression. *Nat. Biotech.* **22**, 841–847. (doi:10.1038/nbt986)

Itaya, M., Tsuge, K., Koizumi, M. & Fujita, K. 2005 Combining two genomes in one cell: stable cloning of the Synechocystis PCC6803 genome in the *Bacillus subtilis* 168 genome. *Proc. Natl Acad. Sci. USA* **102**, 15 971–15 976. (doi:10.1073/pnas.0503868102)

Kitano, H. 2003 A graphical notation for biochemical networks. *Biosilico* **1**, 169–176. (doi:10.1016/S1478-5382(03)02380-1)

Kitano, H., Funahashi, A., Matsuoka, Y. & Oda, K. 2005 Using process diagrams for the graphical representation of biological networks. *Nat. Biotechnol.* **23**, 961–966. (doi:10.1038/nbt1111)

Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I., Osawa, E. & Matsubara, H. 1997 RoboCup: a challenge problem for AI. *AI Mag.* **18**, 73.

Le Novère, N. *et al.* 2005 Minimum information requested in the annotation of biochemical models (MIRIAM). *Nat. Biotechnol.* **23**, 1509–1515. (doi:10.1038/nbt1156)

Le Novère, N. *et al.* 2006 BioModels database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems. *Nucleic Acids Res.* **34**, D689–D691. (doi:10.1093/nar/gkj092)

Le Novère, N. *et al.* 2008 Systems biology graphical notation: process diagram level 1. *Nat. Precedings.* (hdl:10101/npre.2008.2320.1).

Lloyd, C. M., Lawson, J. R., Hunter, P. J. & Nielsen, P. 2008 The CellML model repository. *Bioinformatics* **24**, 2122–2123. (doi:10.1093/bioinformatics/btn390)

Loew, L. M. & Schaff, J. C. 2001 The virtual cell: a software environment for computational cell biology. *Trends Biotechnol.* **19**, 401–406. (doi:10.1016/S0167-7799(01)01740-1)

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**, 1406–1407. (doi:10.1093/bioinformatics/btl086)

Peccoud, J. *et al.* 2008 Targeted development of registries of biological parts. *PLoS ONE* **3**, e2671. (doi:10.1371/journal.pone.0002671)

PricewaterhouseCoopers 2008 Pharma 2020: virtual R7D—which path will you take? See http://www.pwc.com/extweb/pwcpublications.nsf/docid/91BF330647FFA402852572F2005ECC22.

Ramsey, S., Orrell, D. & Bolouri, H. 2005 Dizzy: stochastic simulation of large-scale genetic regulatory networks. *J. Bioinformatics Comput. Biol.* **3**, 415–436. (doi:10.1093/bioinformatics/btm231)

Richardson, S. M., Wheelan, S. J., Yarrington, R. M. & Boeke, J. D. 2006 GeneDesign: rapid, automated design of multikilobase synthetic genes. *Genome Res.* **16**, 550–556. (doi:10.1101/gr.4431306)

Rodrigo, G., Carrera, J. & Jaramillo, A. 2007*a* Asmparts: assembly of biological model parts. *Syst. Synth. Biol.* **1**, 167–170. (doi:10.1007/s11693-008-9013-4)

Rodrigo, G., Carrera, J. & Jaramillo, A. 2007*b* Genetdes: automatic design of transcriptional networks. *Bioinformatics* **23**, 1857–1858. (doi:10.1093/bioinformatics/btm237)

Rullmann, J. A., Struemper, H., Defranoux, N. A., Ramanujan, S., Meeuwisse, C. M. & van Elsas, A. 2005 Systems biology for battling rheumatoid arthritis: application of the Entelos PhysioLab platform. *IEE Proc. Syst. Biol.* **152**, 256–262. (doi:10.1049/ip-syb:20050053)

Sauro, H. M., Hucka, M., Finney, A., Wellock, C. & Bolouri, H. 2003 Next generation simulation tools: the Systems Biology Workbench and BioSPICE integration. *Omics* **7**, 355–372. (doi:10.1089/153623103322637670)

Shapiro, B. E., Levchenko, A., Meyerowitz, E. M., Wold, B. J. & Mjolsness, E. D. 2003 Cellerator: extending a computer algebra system to include biochemical arrows for signal transduction simulations. *Bioinformatics* **19**, 677–678. (doi:10.1093/bioinformatics/btg042)

Shetty, R. P., Endy, D. & Knight, T. 2008 Engineering BioBrick vectors from BioBrick parts. *J. Biol. Eng.* **2**, 5. (doi:10.1186/1754-1611-2-5)

Simpson, M. L. 2006 Cell-free synthetic biology: a bottom-up approach to discovery by design. *Mol. Syst. Biol.* **2**, 69. (doi:10.1038/msb4100104)

Stricker, J., Cookson, S., Bennett, M. R., Mather, W. H., Tsimring, L. S. & Hasty, J. 2008 A fast, robust and tunable synthetic gene oscillator. *Nature* **456**, 516–519. (doi:10.1038/nature07389)

Strömbäck, L., Hall, D. & Lambrix, P. 2007 A review of standards for data exchange within systems biology. *Proteomics* **7**, 857–867. (doi:10.1002/pmic.200600438)

The Economist 2005 Models that take drugs. *The Economist Report*, 11 June 2005.

Visualization of omics data for systems biology

Nils Gehlenborg[1,2], Seán I O'Donoghue[3], Nitin S Baliga[4], Alexander Goesmann[5], Matthew A Hibbs[6], Hiroaki Kitano[7–9], Oliver Kohlbacher[10], Heiko Neuweger[5], Reinhard Schneider[3], Dan Tenenbaum[4] & Anne-Claude Gavin[3]

High-throughput studies of biological systems are rapidly accumulating a wealth of 'omics'-scale data. Visualization is a key aspect of both the analysis and understanding of these data, and users now have many visualization methods and tools to choose from. The challenge is to create clear, meaningful and integrated visualizations that give biological insight, without being overwhelmed by the intrinsic complexity of the data. In this review, we discuss how visualization tools are being used to help interpret protein interaction, gene expression and metabolic profile data, and we highlight emerging new directions.

Visualization has long been key in helping to understand biological systems, such as metabolism[1], signaling[2] and the regulation of gene expression[3]. In recent years, the study of such systems has been profoundly influenced by the development of a wide range of high-throughput experimental methods (**Box 1**), resulting in a greatly increased volume of complex, interconnected data. Remarkably, in spite of these changes, and in spite of the development of new methods for visualizing and analyzing these data, we still use the same primary visual metaphor to communicate ideas about biological systems: namely, pathways (graphs that show overall changes in state) or, more generally, networks (graphs that do not necessarily show state changes).

As high-throughput experimental methods have become more routine, many more scientists are using network and pathway visualization to record and communicate their findings. There are now over 300 web resources[4] (see http://pathguide.org/) providing access to many thousands of pathways and networks that document millions of interactions between proteins, genes and small molecules.

There has been a corresponding increase in the development of visualization tools for systems biology data[5–7]. These tools are very diverse, but they can be broadly divided into two partly overlapping categories, the first consisting of tools focused on automated methods for interpreting and exploring large biological networks (**Table 1**), and the second consisting of tools focused on assembly and curation of pathways (**Table 2**). Many of these tools are tightly integrated with public databases, thus allowing users to visualize and interpret their own data in the context of previous knowledge.

For users and developers of these visualization tools, one of the key challenges is how to benefit from the explosion in systems biology data without being overwhelmed by it—or, in practical terms, how to present the data at the right level of detail, in a cohesive, insightful manner. Clearly, the answers depend on context.

In this review, we first discuss the methods and tools now being used to visualize and analyze data sets from three main types of high-throughput experiments: namely, the investigation of protein-protein interactions, of gene expression profiles and of metabolic profiles. Such experiments are used to study cellular response to a wide variety of conditions—including drug exposure, disease states and specific genetic

[1]European Bioinformatics Institute, Cambridge, UK. [2]Graduate School of Life Sciences, University of Cambridge, Cambridge, UK. [3]European Molecular Biology Laboratory, Heidelberg, Germany. [4]Institute for Systems Biology, Seattle, Washington, USA. [5]CeBiTec, Bielefeld University, Bielefeld, Germany. [6]The Jackson Laboratory, Bar Harbor, Maine, USA. [7]Sony Computer Science Laboratories, Tokyo, Japan. [8]The Systems Biology Institute, Tokyo, Japan. [9]Okinawa Institute of Science and Technology, Okinawa, Japan. [10]University of Tübingen, Tübingen, Germany. Correspondence should be addressed to S.I.O. (sean.odonoghue@embl.de).

PUBLISHED ONLINE 1 MARCH 2010; DOI:10.1038/NMETH.1436

BOX 1 KEY EXPERIMENTAL METHODS FOR SYSTEMS BIOLOGY

Oligonucleotide microarrays. The most widely used methods to monitor the expression levels of RNA transcripts in a biological sample are based on microarrays. They measure the hybridization of fluorescently labeled cDNA, synthesized from extracted mRNA, to known nucleotide sequences spotted on solid surfaces[117]. For all genes on the microarray, an expression value is derived from the fluorescence intensity of the hybridized RNAs. These expression values are typically unitless and have meaning only in the context of a reference measurement. Before further analysis takes place, the measurements must therefore be normalized to remove systematic biases and to make it possible to compare measurements from different samples.

Quality assessment is likewise essential for the validity of later analyses. This is typically performed with the help of (platform-dependent) quality scores at the level of both individual probes and entire arrays, complemented by diagnostic visualization tools that have been developed for this purpose[118,119]. Evaluating the quality of individual arrays is routinely done with spatial intensity distributions plots and plots of intensity ratio versus mean intensity (**Supplementary Fig. 1a**). Comparison of multiple arrays can be achieved with intensity box plots, which are a practical tool to detect outlier arrays that should be excluded from subsequent analysis (**Supplementary Fig. 1b**). Several tools that provide quality assessment visualizations are listed in **Supplementary Table 1**.

RNA deep sequencing. The most recent transcriptomics approaches are based on the deep sequencing of transcripts extracted from biological samples[33]. The resulting sequence reads—typically 30 to 400 base pairs long, depending on the DNA-sequencing technology used—are then commonly aligned to a reference genome and evaluated to determine their quality.

Tools for data processing and quality assessment typically provide diagnostic visualizations. Examples include the R/Bioconductor packages ShortRead[120] and edgeR[121]. The latter provides many functions that are analogous to those in the limma package[122] for transcriptomics data from microarrays. Reads aligned to a genome can also be visualized and evaluated with some of the more recent genome browsers that can handle short read data, such as the Integrative Genomics Viewer (http://www.broadinstitute.org/igv/). This and similar tools are discussed in the accompanying review by Nielsen *et al.*[72].

Mass spectrometry. In mass spectrometry (MS) experiments, the compounds present in a sample are identified through the accurate measurements of their mass-to-charge ratios. MS has applications in many fields, including proteomics, metabolomics and interactome mapping.

In proteomic applications, typical MS data sets consist of lists of proteolytic peptides characterized by their mass-to-charge ratios (MS spectra, MS1). These peptides can be further fragmented and measurements of the resulting mass spectra (MS-MS spectra or tandem MS spectra, MS2) used to deduce their sequences. In some cases, complex samples must be fractionated and proteolytic peptides are separated using high performance liquid chromatography (LC) before MS analysis (LC-MS).

Several search engines have been developed to predict peptides and proteins through the comparison of experimentally measured spectra to theoretical spectra (predicted from sequence databases). Quality scores provide a measure of the reliability of a given protein or peptide identification[123]. For example, for Mascot[124], the most broadly used algorithm, the score features the number of identified peptides (sequence coverage).

The overall quality of entire MS data sets is generally measured by the false discovery rate (FDR), which is the 'expected' proportion of incorrect assignments among the accepted assignments. The most popular approach to calculate FDR is based on the use of a target-decoy database[123]. Also, an array of visualization tools has been developed to evaluate the technical quality of the samples and of MS runs. For example, the overall distribution of peptides in an LC-MS map can be visualized with Pep3D[125] or TOPPView[126], enabling the detection of possible biases, the presence of chemical contaminants or poor separations during the LC (**Supplementary Fig. 2**). Additionally, Pep3D can integrate quality scores for individual protein or peptide identifications generated by search engines into these maps (**Supplementary Fig. 3**). We list tools for mass spectrometry data visualization and evaluation in **Supplementary Table 1**.

In metabolomics applications, owing to large chemical diversity and variation in molecular composition of the analytes, various chromatographic systems, such as gas chromatography (GC), LC or electrochemistry (EC), are generally applied before MS. GC-MS is the most popular method for global metabolite profiling[127]. It can be complemented with LC-MS analysis to identify compounds that are not suitable for GC-MS analysis[128]. Similarly to the approaches developed for peptides, metabolites can be identified on the basis of their fragmentation patterns, for which mass spectral fingerprint libraries are being developed. Because the raw data are of the same kind as in proteomics mass spectrometry studies, very similar visualization methods are used to assess data quality (**Supplementary Table 1**).

Nuclear magnetic resonance. Nuclear magnetic resonance (NMR) is a common method in metabolomics and, in contrast to MS-based approaches, in most cases does not require analyte separation. NMR spectroscopy can provide detailed information on the molecular structure of compounds found in complex mixtures, and a wide range of small molecule metabolites in a sample can be detected simultaneously. Biofluids, cell and tissue extracts can be analyzed with minimal sample preparation through the use of ^{1}H NMR spectroscopy[129]. With the use of two-dimensional NMR spectra, the identification and reliable quantification of individual metabolites becomes feasible, which enables NMR-based metabolite profiling. Data processing and spectral deconvolution are challenging, and databases of NMR spectra of pure metabolites are not yet comprehensive, but they nonetheless do already help in the identification process[130]. Applications such as MetaboMiner[131] can be used for the semiautomated identification of metabolites in two-dimensional NMR spectra, supported by visualizations that allow the scientist to inspect the matches of peaks to reference spectra and assess match quality.

perturbations (for example, gene deletions, gene insertions and siRNA knockdowns). Often, these experiments produce new knowledge that is then either added to existing pathways or used to create new pathways. Thus, we end with a discussion of methods and tools for pathway editing.

Protein interaction data

A range of experimental methods are at present being used for high-throughput studies of protein interactions[8]. For instance, in yeast, pairwise interactions have been studied on the genome scale using yeast two-hybrid screens or protein complementation assays, whereas the assembly of proteins within complexes has been systematically charted using tandem affinity purification coupled with mass spectrometry (TAP-MS) (**Box 1** and **Supplementary Figs. 2** and **3**). Recent analyses have estimated that, in yeast, some 20,000 pairwise interactions may take place between the ~5,000 gene products[9], and about 800 protein complexes may exist[10]. As a result of these and similar studies in other species, vast amounts of protein interaction data are accumulating in public databases[11,12] such as DIP[13], HPRD[14] and IntAct[15].

The size and complexity of these data sets can be daunting; hence, a common general strategy is to iteratively dissect the data sets into smaller subsets. Typically, these subsets are defined as sets of proteins that belong to the same complex, or that are found at the same subcellular location, or that belong to a similar functional category. Visualization is key in this strategy, as human judgment and intervention are often needed, in part because of errors (false positives and false negatives) in protein interaction data sets[9,16]. Many visualization tools have been developed specifically to support the analysis of protein networks (**Table 1**); here we discuss how these tools can be used to help dissect large data sets of interactions, extract biological insight and generate hypotheses leading to further experimental investigations.

As proteins rarely act alone, a first step in analyzing a protein interaction data set is to identify protein complexes and groups of complexes. For small, simple networks, visualized as a graph in which each node represents a protein and each edge represents an interaction between two proteins, the arrangement of proteins and complexes can usually be seen clearly using a standard 'force-directed' layout[17], which automatically arranges each node

Table 1 | Visualization tools focused on interaction networks

Name	Cost	OS	Description	URL
Stand-alone				
Arena 3D[63]	Free	Win, Mac, Linux	Visualization of biological multi-layer networks in 3D	http://www.arena3d.org/
BiNA[81]	Free	Win, Mac, Linux	Exploration and interactive visualization of pathways	http://www.bnplusplus.org/bina/
BioLayout Express 3D[37]	Free	Win, Mac, Linux	Generation and cluster analysis of networks with 2D/3D visualization	http://www.biolayout.org/
BiologicalNetworks[82]	Free	Win, Mac, Linux	Analysis suite; visualizes networks and heat map; abundance data	http://www.biologicalnetworks.org/
Cytoscape*[20,83]	Free	Win, Mac, Linux	Network analysis; extensive list of plug-ins for advanced visualization	http://www.cytoscape.org/
GENeVis[36]	Free	Win, Mac, Linux	Network and pathway visualization; abundance data	http://tinyurl.com/genevis/
Medusa[84]	Free	Win, Mac, Linux	Basic network visualization tool	http://coot.embl.de/medusa/
N-Browse[85]	Free	Win, Mac, Linux	Network visualization software for heterogeneous interaction data	http://www.gnetbrowse.org/
NAViGaTOR[23,86]	Free	Win, Mac, Linux	Visualization of large protein-interaction data sets; abundance data	http://tinyurl.com/navigator1/
Ondex[87]	Free	Win, Mac, Linux	Integrative workbench: large network visualizations; abundance data	http://www.ondex.org/
Osprey[88]	Free	Win, Mac, Linux	Tool for visualization of interaction networks	http://tinyurl.com/osprey1/
Pajek[89]	Free	Win	Generic network visualization and analysis tool	http://pajek.imfm.si/
ProViz	Free	Win, Mac, Linux	Software for visualization and exploration of interaction networks	http://tinyurl.com/proviz/
SpectralNET[90]	Free	Win	Network visualizations; scatter plots for dimensionality reduction methods	http://tinyurl.com/spectralnet/
Tulip[91]	Free	Win, Mac, Linux	Generic visualization tool; extremely large networks; 3D support	http://tulip.labri.fr/TulipDrupal/
VANTED[21]	Free	Win, Mac, Linux	Combined visualization of abundance data, networks and pathways	http://tinyurl.com/vanted/
yEd	Free	Win, Mac, Linux	Generic network visualization software; offers many layout algorithms	http://tinyurl.com/yEdGraph/
Cytoscape plug-in				
BiNoM[92]	Free	Win, Mac, Linux	Extensive support for common systems biology network formats	http://tinyurl.com/binom1/
BioModules[24]	Free	Win, Mac, Linux	Detects modules in networks; maps abundance data onto nodes and modules	http://tinyurl.com/biomodules/
Cerebral*[26,78]	Free	Win, Mac, Linux	Biologically motivated layout algorithm; maps abundance data; clustering	http://tinyurl.com/cerebral1/
MCODE[18]	Free	Win, Mac, Linux	Network clustering algorithm; support for manual cluster refinement	http://tinyurl.com/MCODE123/
VistaClara[42]	Free	Win, Mac, Linux	Mapping of abundance data to nodes and 'heat strips'; provides heat map	http://tinyurl.com/cytoplugins/
Web-based				
Graphle[93]	Free		Distributed client/server network exploration and visualization tool	http://tinyurl.com/graphle/
Lichen	Free		Library for web-based visualization of network and abundance matrix data	http://tinyurl.com/Lichen1/
MAGGIE Data Viewer	Free		Visualization of networks; abundance data in heat maps and profile plots	http://maggie.systemsbiology.net/
STITCH[31]	Free		Construction and visualization of networks from a wide range of sources	http://stitch.embl.de/
VisANT[22]	Free	Win, Mac, Linux	Analysis, mining and visualization of pathways and integrated omics data	http://visant.bu.edu/

Some of the tools in this table have capabilities similar to tools that are listed in other tables. To avoid listing tools in more than one table, we assigned tools to tables on the basis of what we understand to be their primary purpose. *Our recommendations. Free means the tool is free for academic use; $ means there is a cost. OS, operating system: Win, Microsoft Windows; Mac, Macintosh OS X. Tools running on Linux usually also run on other versions of Unix. 2D, two-dimensional; 3D, three-dimensional.

to minimize the number of edge crossings while trying to keep the lengths of all edges approximately the same. However, this approach quickly becomes inadequate as the network size and complexity increase (**Fig. 1a**). Instead, clustering approaches are used, which predict higher-order protein complexes from the interaction data. One very commonly used tool for this purpose is MCODE[18]. For TAP-MS and other data sets where components of protein complexes are experimentally determined, other clustering methods are used (for example, 'clique percolation'[19]). The results of these clustering analyses can then be used to change the layout and appearance of the network (**Fig. 1a,b**) in a way that may yield biological insights that cannot be easily obtained by simply examining lists of proteins or protein complexes. For instance, by viewing the network, the scientist may notice connections between two complexes that suggest a previously unknown biological relationship. Furthermore, on the basis of previous knowledge, the scientist may be able to assign a putative function or subcellular localization to the complex; this information can be visualized using node color or shape to represent the functional category or location of the proteins. Similarly, node color or shape can be used to show which proteins belong to the same complex (**Fig. 1b**).

Most network visualization tools provide the ability to interactively change the layout of the network—for example, by automatically arranging a user-defined group of proteins into any of a variety of arrangements (a circle, a line and so forth) or by manually moving nodes. This ability can be very useful in creating visualizations that emphasize biologically significant relationships and interactions between complexes (**Fig. 1c**) or between 'hub' proteins and their partners (for example, between kinases and their substrates). Tools that support such interactive editing particularly well include Cytoscape[20], VANTED[21], VisANT[22] and NAViGaTOR[23].

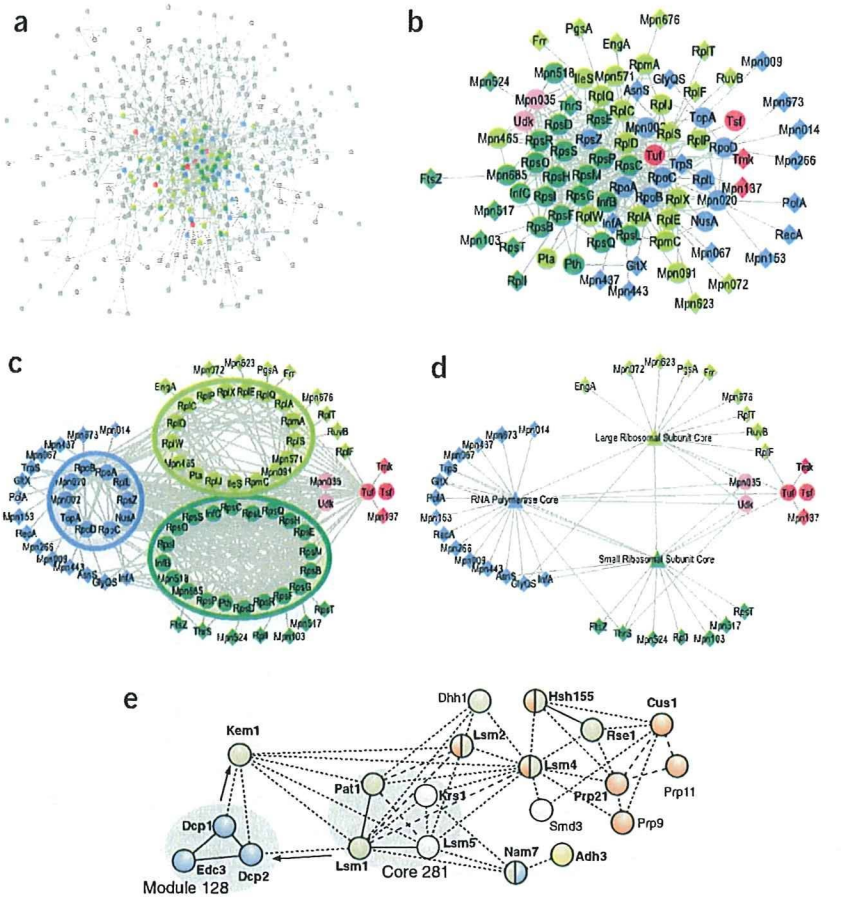

Figure 1 | Visualization of protein interaction networks. (**a–d**) Cytoscape[20] images of *Mycoplasma pneumoniae* protein interaction data derived by mass spectrometry[19] analysis. (**a**) Initial protein interaction network (>400 proteins) laid out with a force-directed algorithm. Nodes discussed in the following steps are overlaid with functional annotations (blue, RNA polymerase; dark or light green, small or large ribosomal subunits, respectively; red, elongation factor). (**b**) Recomputed network remaining after removal of nodes not of interest. Five computationally determined complexes are colored according to functional annotation. Node shapes represent different roles in the complex (circle, core protein of complex; diamond, protein attached to complex but not part of the core). At this stage, clusters emerge. (**c**) Manual refinement of the network layout emphasizing structure of protein complexes and interactions between them. (**d**) Collapse of nodes in each complex core, simplifying the network and emphasizing global properties. (**e**) Stages in deadenylation-dependent mRNA degradation in *Saccharomyces cerevisiae*. Reproduced from Gavin *et al.*[10]. Arrows show the order of sequential steps in a cellular process. Proteins are colored according to their localization (green, cytoplasm; red, nucleus; blue, punctate composite (undefined subcellular structure); yellow, mitochondria; white, unknown). Edge styles represent socio-affinity indices (dotted, 5–10; dashed, 10–15; solid, >15). TAP-MS bait proteins, bold; shaded circles, protein complexes.

It is often useful to collapse all members of a protein complex or cluster into a single 'meta-node' (**Fig. 1d**) that can later be expanded, depending on screen space and the desired level of detail. Meta-nodes not only simplify the appearance of the network, they can also be useful in more clearly illustrating biological relationships between protein complexes. Meta-nodes can also help to visually arrange the network to give insight into the integration and coordination of cellular functions (**Fig. 1d**). Meta-nodes are supported by yEd (http://tinyurl.com/yEdGraph/), BioModules[24] and VisANT, the last of which further allows meta-nodes to be nested hierarchically and can show 'meta-edges'[25] between meta-nodes—these can indicate, for example, when proteins are shared between two collapsed complexes.

Present high-throughput experimental methods often do not determine the spatial, or subcellular, location where an interaction takes place, so it can be highly informative to include any previous protein localization information in the analysis of these data sets. For instance, the network may be filtered to show only proteins known to occur in selected locations, thus simplifying it and allowing the scientist to focus only on interactions within a defined subcellular location. Alternatively, subcellular location

Table 2 | Visualization tools focused on pathways

Name	Cost	OS	Description	URL
Stand-alone				
BioTapestry[94]	Free	Win, Mac, Linux	Visualization of genetic regulatory networks, also with experimental data	http://www.biotapestry.org/
Caleydo[95]	Free	Win, Linux	Interactive framework for pathway and expression data; 3D 'bucket' view	http://www.caleydo.org/
CellDesigner*[51]	Free	Win, Mac, Linux	Drawing and simulation of pathways and models; supports SBGN	http://www.celldesigner.org/
Edinburgh Pathway Editor	Free	Win, Mac, Linux	Construction and visualization of pathway diagrams; supports SBGN	http://tinyurl.com/EdinburghPE/
GenMAPP[40]	Free	Win	Pathway visualization and construction; abundance data	http://www.genmapp.org/
IngenuityPathways	$	Win, Mac, Linux	Full analysis suite; network and pathway visualizations; abundance data	http://tinyurl.com/IngenuityPath/
JDesigner[52]	Free	Win	Drawing and simulation of pathways and models	http://tinyurl.com/jdesigner/
KaPPA View[48]	Free	Win	Analysis and visualization of plant pathways and mapped abundance data	http://tinyurl.com/kappa-view/
KEGG Atlas[96]	Free	Win, Mac, Linux	Visualization of abundance data on interactive KEGG pathways	http://www.genome.jp/kegg/
MetaCore	$	Win, Mac, Linux	Pathway, network and omics data analysis and visualization suite	http://www.genego.com/
PathVisio[97]	Free	Win, Mac, Linux	Pathway visualization and editing; supports mapping of omics data	http://www.pathvisio.org/
VitaPad[98]	Free	Win, Mac, Linux	Editing of pathway diagrams; integration of abundance data	http://tinyurl.com/vitapad/
Web-based				
ArrayXPath[99]	Free		Mapping of abundance data to pathway visualizations	http://tinyurl.com/ArrayXPath/
GEPAT[100]	Free		Analysis suite; visualization of transcriptomics data on pathways maps	http://tinyurl.com/GEPAT1/
iPath[101]	Free		Visualization and exploration of combined KEGG pathways	http://pathways.embl.de/
MapMan[46]	Free		Visualization of abundance data on metabolic pathways	http://tinyurl.com/MapManApp/
Omics Viewer[47,102]	Free		Mapping of abundance data to BioCyc pathway diagrams	http://www.biocyc.org/
Pathway Explorer[49]	Free		Visualization of abundance data on pathways	http://tinyurl.com/pathwayexp/
PATIKA[103]	Free		Pathway visualization suite; good support for signaling pathways	http://www.patika.org/
Payaologue	Free		Collaborative pathway annotation and visualization tool	http://celldesigner.org/payao/
ProMeTra[41]	Free		Maps abundance matrices of multiple data types to pathways	http://tinyurl.com/ProMeTra/
Reactome SkyPainter[30]	Free		Visualization of over-represented pathways and reactions from gene lists	http://reactome.org/
WikiPathways[62]	Free		Wiki-based, community-driven pathway curation and visualization tool	http://www.wikipathways.org/

Some of the tools in this table have capabilities similar to tools that are listed in other tables. To avoid listing tools in more than one table, we assigned tools to tables on the basis of what we understand to be their primary purpose. *Our recommendations. Free means the tool is free for academic use; $ means there is a cost. OS, operating system: Win, Microsoft Windows; Mac, Macintosh OS X. Tools running on Linux usually also run on other versions of Unix. 3D, three-dimensional.

can be indicated using node coloring; this can be particularly useful when studying the interactions of complexes that move between subcellular locations (**Fig. 1e**). Another common strategy is to arrange the network so that all proteins belonging to the same subcellular location are gathered together in one region (**Fig. 2, see 2a**). For small networks, such a layout depicting subcellular localization is often created manually. However, for large networks, it is much more convenient to use tools that can achieve such a layout automatically, such as Cerebral[26] (**Fig. 2a**) and PATIKA[27]. These tools also draw boundaries or use shading so that the scientist can see clearly which regions of the network correspond to which subcellular locations.

Protein interaction data sets commonly do not capture information about dynamic changes in protein abundance. Thus, as with spatial information, it is often useful to include temporal information from other experiments—for example, by identifying proteins whose abundance is known to vary throughout the cell cycle. This information can be used to simplify a large network by either depicting only proteins that are coexpressed or by mapping expression or abundance profiles of proteins of interest onto nodes[28] in the network, as described in more detail below (see Network enrichment).

These processes of dissection are all aimed at dividing a protein interaction data set into manageable, biologically significant parts that can be interpreted; during this process of interpretation, a scientist often makes use of previously established knowledge, particularly pathways (for example, KEGG[29] or Reactome[30]) and networks (for example, STITCH[31]). In some cases, to illustrate a result or insight, it can be useful to add interactions derived from previous studies—thus forming a hybrid network that shows both new and old data (**Fig. 1e**).

Expression profile data

A range of experimental methods are being used for high-throughput expression profiling (**Box 1, Supplementary Fig. 1** and **Supplementary Table 1**); in addition to gene expression profiling with DNA microarrays[32] and RNA deep sequencing[33], a promising emerging technology is quantitative protein expression profiling based on mass spectrometry[34,35]. Gene expression profile data sets are being deposited in two main repositories, ArrayExpress (http://www.ebi.ac.uk/arrayexpress/) and Gene Expression Omnibus (http://www.ncbi.nlm.nih.gov/geo/), with around 15,000 studies now in the public domain.

The initial goal in analyzing expression profiles is usually to find a set of genes or, less typically, proteins that share a related pattern of expression—for example, genes that are up- or down-regulated in a certain genotype, disease model or human disease, or in response to a drug treatment. The challenge is that a single data set may contain expression profiles for over 10,000 genes, measured over a range of time points and experimental conditions, so that determining which genes are potentially relevant to the studied problem requires an extensive search through a large amount of often noisy, multivariate data. Together with various clustering algorithms, visualization is key in these analyses[32], and

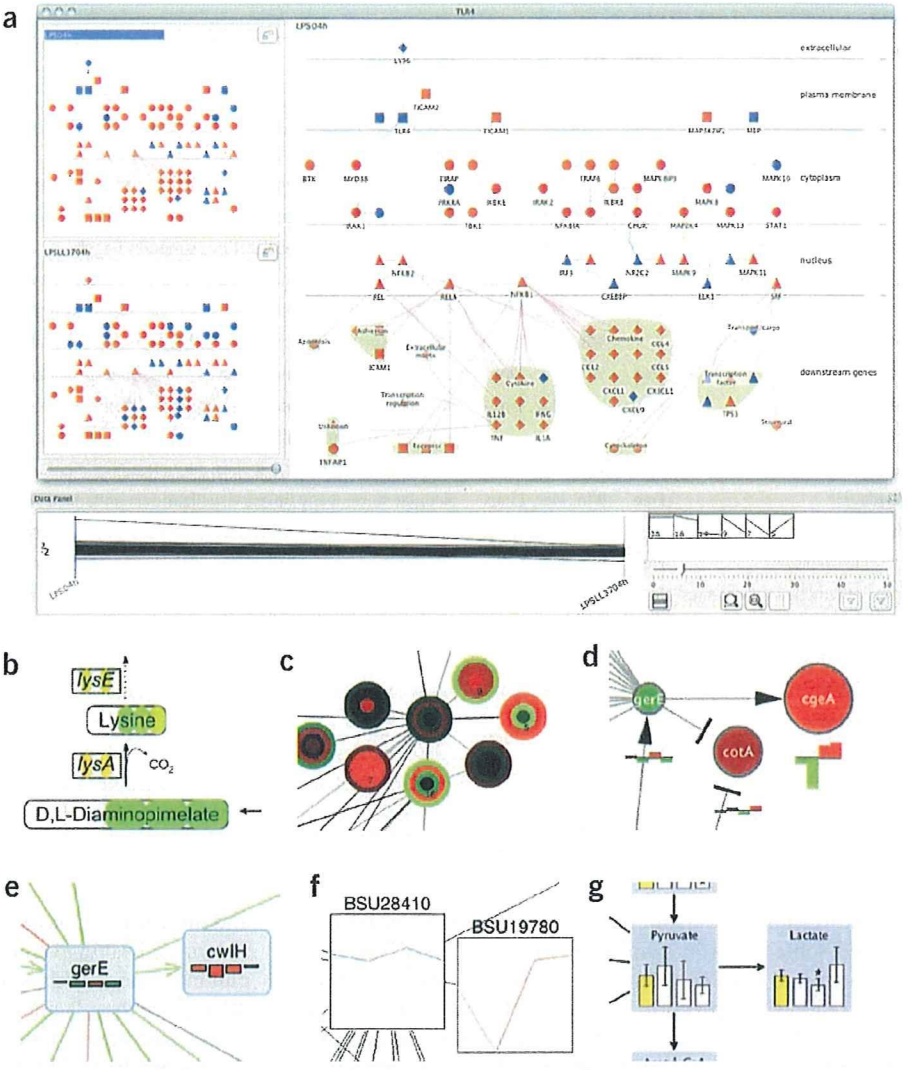

Figure 2 | Omics data overlaid onto biological networks. (**a**) Cerebral[78] showing the TLR4-to-NF-κB signaling pathway[79] laid out according to subcellular localization and functional annotation (green shading). Direction of information flow is from top to bottom. Node colors represent relative expression (red, upregulation; blue, downregulation) and edge colors represent interaction type (orange, phosphorylation; cyan, other protein interaction; purple, transcriptional regulation). The left two panels are a 'small multiples' display of the same pathway overlaid with gene expression data for two experimental conditions. In this case, the top panel has been selected, and hence is also shown in the main window. The bottom panels show the detailed expression profiles corresponding to genes shown in the pathway panel (see 'Network enrichment'). The data set shows how upregulation of NFKB1 explains the observed upregulation of several chemokine proteins. (**b**) ProMeTra[41] display showing both metabolomics and transcriptomics time series data from five time points. Metabolite and enzyme nodes in the pathway map are subdivided into five areas, one per time point. Areas are color coded (green, upregulation; yellow, no change; white, missing data) to indicate metabolite concentrations and transcript levels relative to a reference time point. (**c**) Lichen rendering of a gene regulatory network overlaid with transcriptomics data using a circular heat map. Each concentric ring represents a time point, and the color of the circle represents expression level (red, upregulation; green, downregulation; black, no change). Numbers identify genes. (**d**) VistaClara[42] display showing transcript levels across four time points relative to a reference time point as 'heat strips' below the nodes (height of bar, relative expression; red, upregulation; green, downregulation). Node color indicates expression level from time point 4. Node size indicates reliability of the measurement taken at time point 4. (**e**) GENeVis[36] visualization of the same data set. Color coding as in **d**; height of bar corresponds directly to the reliability of the measurement at each time point. (**f**) Profile plots of the same data embedded in nodes in VisANT[22]. The color of each line segment represents the change in expression levels between two time points (red and purple, increase; blue and cyan, decrease). (**g**) Visualization of metabolite concentrations in a pathway map in VANTED[21] using a bar chart with error bars.

a wide range of tools have been developed to aid the visualization process (**Table 3**). Many of these tools implement a set of commonly applied methods (**Box 2** and **Fig. 3**); in particular, scatter plots combined with dimensionality reduction (**Fig. 3a**), profile plots (**Fig. 3b**), heat maps, and dendrograms (**Fig. 3c**), as well as clustering. As microarray gene expression analysis has matured as an experimental technique, many of the corresponding visualization methods have become well established and are widely used.

Network enrichment. Once a list of potentially relevant genes has been found using the above types of analysis, the next task is often to find pathways or networks where these genes are significantly over-represented. These 'enrichment' searches can be launched directly from several network visualization tools, for example, GENeVis[36], Reactome SkyPainter[30], Metacore (GeneGo Inc.) or BioLayout Express 3D[37]. A logical next step is then to map gene expression levels onto the identified pathways. Interpreting expression data in the context of a visualized pathway or network usually proves more insightful than without this type of information. For instance, visualizing the data in the context of pathways may show how the upregulation of a transcription factor explains the upregulation of many other genes under its control (**Fig. 2a**) and may lead to testable experimental hypotheses.

A wide range of representations are used for mapping gene expression levels onto pathways and networks, with the ideal choice depending on the specific experiment and question of interest[38] (**Fig. 2**). A simple approach that is available as part of many tools (**Table 1**) is to represent expression levels as a color gradient, as in a heat map, and then color the nodes in the network according to their expression level under a particular condition (**Fig. 2a**).

Table 3 | Visualization tools for multivariate omics data

Name	Cost	OS	Description	URL
Stand-alone				
BicOverlapper[104]	Free	Win, Mac, Linux	Visualization of biclusters combined with profile plots and heat maps	http://vis.usal.es/bicoverlapper/
BiGGEsTS[105]	Free	Win, Mac, Linux	Heat map–based bicluster visualization	http://tinyurl.com/BiGGEsTS/
Brain Explorer[76]	Free	Win, Mac	Visualization of 3D transcription data in the central nervous system	http://tinyurl.com/brainExplorer/
Caryoscope[75]	Free	Win, Mac, Linux	Abundance data mapped to chromosomal location	http://tinyurl.com/caryoscope/
Data Matrix Viewer	Free	Win, Mac, Linux	Simple profile plot visualization; supports Gaggle	http://gaggle.systemsbiology.net/
EXPANDER[106]	Free	Win, Linux	Heat maps, scatter plots and profile plots of cluster averages	http://acgt.cs.tau.ac.il/expander/
Genesis[107]	Free	Win, Mac, Linux	Analysis suite; offers several interactive visualizations	http://tinyurl.com/genesisclient/
GeneSpring GX*	$	Win, Mac, Linux	Analysis suite; interactive and linked visualizations; also networks	http://tinyurl.com/genespring/
GeneVAnD[108]	Free	Win, Mac, Linux	Linked heat maps, dendrograms and 2D/3D scatter plots	http://tinyurl.com/GeneVAnD/
geWorkbench	Free	Win, Mac, Linux	Modular suite; heat maps, dendrograms, profile and scatter plots	http://tinyurl.com/geWorkbench/
HCE*[109]	Free	Win	Linked heat map, profile and scatter plots; systematic exploration	http://tinyurl.com/HCExplorer/
Java TreeView*[110]	Free	Win, Mac, Linux	Linked heat maps, karyoscopes, sequence alignments, scatter plots	http://jtreeview.sourceforge.net/
Mayday[111]	Free	Win, Mac, Linux	Modular suite; many linked visualizations; enhanced heat map[112]	http://tinyurl.com/maydaywp/
MultiExperiment Viewer*[113]	Free	Win, Mac, Linux	Analysis suite; heat maps, dendrograms, profile and scatter plots	http://www.tm4.org/
PointCloudXplore[77]	Free	Win, Mac, Linux	Visualization of 3D transcription data in *Drosophila* embryos	http://tinyurl.com/PointCloudXplore/
Spotfire Functional Genomics	$	Win	Analysis suite; many linked visualizations and exploration tools	http://spotfire.tibco.com/
TimeSearcher[114]	Free	Win	Exploration and analysis of time series; advanced profile plots	http://tinyurl.com/timesearcher/
R/BioConductor				
Geneplotter	Free	Win, Mac, Linux	Karyoscope-style plots and other visualizations	http://www.bioconductor.org/
Web-based				
ExpressionProfiler[115]	Free		Transcriptomics data analysis suite with basic visualizations	http://tinyurl.com/expresspro/
GenePattern[116]	Free		Modular analysis platform; several visualization modules available	http://tinyurl.com/GenePatt/

Some of the tools in this table have capabilities similar to tools that are listed in other tables. To avoid listing tools in more than one table, we assigned tools to tables on the basis of what we understand to be their primary purpose. *Our recommendations. Free means the tool is free for academic use; $ means there is a cost. OS, operating system: Win, Microsoft Windows; Mac, Macintosh OS X. Tools running on Linux usually also run on other versions of Unix. 2D, two-dimensional; 3D, three-dimensional.

If expression levels from more than one condition are being studied, some tools (for example, VisANT and VistaClara) allow the scientist to visualize them sequentially, by updating node colors to reflect the expression levels of a selected condition. Some tools switch automatically to depiction of the next condition after a predefined time interval, which leads to an animation-like visualization that is well suited to interpreting data from a time series. An alternative strategy to viewing the data from different conditions in series is to view them in parallel, by arranging multiple versions of the same network in a grid, where each version represents the expression levels (visualized as node color) for one condition or time point. This approach is known as 'small multiples'[39] and allows the scientist to visually compare expression levels between conditions, which is not well supported by animation. A well-designed implementation of small multiples is available in Cerebral[26] (**Fig. 2a**).

Besides animation and small multiples, a third approach is to show the complete expression profile within the nodes of a network. The most common representation of this type is based on a miniature heat map embedded in each node (**Fig. 2b**) and is available in several tools, including GenMAPP 2 (ref. 40), GeneSpring GX (Agilent Technologies) and ProMeTra[41]. The Lichen package (http://tinyurl.com/Lichen1/) uses a circular heat map to depict this information, which has the advantage of being very compact (**Fig. 2c**). VistaClara[42] provides 'heat strips', in which the heights of the bars as well as their colors correspond to the expression levels (**Fig. 2d**). In contrast, in GENeVis, bar heights represent confidence measures, so that reliable measurements are taller and therefore emphasized (**Fig. 2e**). A less common alternative to showing a heat map embedded in the node is to embed a profile plot in the node. This is supported by, for instance, VisANT (**Fig. 2f**) and has the advantage that multiple profiles can be displayed in the same node—for example, when a meta-node represents a set of genes. In VANTED, each node of the network has embedded visualizations of exceptionally high detail, showing legends, grid lines, bar charts or error bars (**Fig. 2g**). Although powerful, this representation requires the nodes to be rather large in order to show the details of the embedded visualizations, which effectively limits its application to only small pathways and networks.

For expression profiles with many conditions, visualizing all these data directly in the network is invariably problematic because of a lack of space, and an approach that links visualization of the network to a separate visualization of the expression profiles is more appropriate. In the linked approach, a heat map (as implemented in VistaClara) or a profile plot (as implemented in Cerebral; **Fig. 2a**) is shown next to the network and when the scientist selects nodes in the network, the corresponding expression profiles are highlighted in the linked heat map or profile plot, or vice versa. This approach allows the scientist to check, for instance, whether the members of a putative protein complex in the network visualization are coexpressed, by comparing the corresponding gene expression profiles in the linked heat map. Conversely, selection of a set of coexpressed genes in a clustered heat map would allow exploration of their role in the linked protein interaction network: the scientist could directly see whether these genes are part of the same complex, what their interaction

BOX 2 KEY VISUALIZATION METHODS FOR MULTIVARIATE DATA

Multivariate data, for instance from gene expression studies, are very common in systems biology, and many tools have been developed to analyze and visualize such data (**Table 3**). The three most commonly used visualization methods are scatter plots, profile plots and heat maps.

Scatter plots. Scatter plots (**Fig. 3a**) are primarily used to examine dependencies between two variables, but in combination with dimensionality reduction methods, they can also be applied to multivariate data. For instance, to gain insight into the global patterns in a gene expression matrix, a dimensionality reduction method may be applied to obtain a two-dimensional (sometimes three-dimensional) representation of the expression profiles, which are then visualized in a scatter plot to reveal clusters and outliers in the data. Some frequently applied dimensionality reduction methods for this purpose are principal component analysis[132] (PCA) and multi-dimensional scaling[133] (MDS), which are implemented in many tools. Besides PCA and MDS, many other suitable dimensionality reduction methods exist[134], but they are often not easily accessible to the casual user.

Scatter plots combined with dimensionality reduction methods are an excellent tool for gaining insight into the overall structure of large sets of expression profiles. However, because of the dimensionality reduction itself, it is not possible to extract information about the relationship between expression levels and the conditions under study.

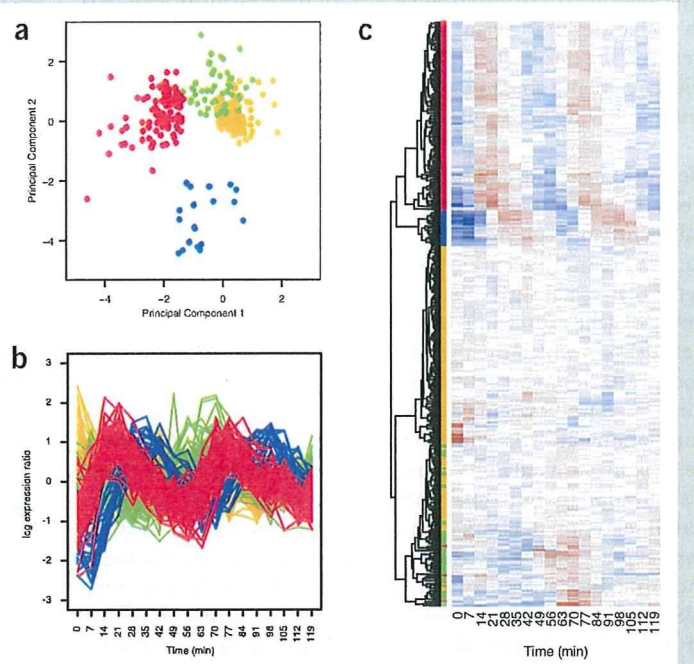

Figure 3 | Visualization of gene expression profiles. Expression of 320 transcripts from *S. cerevisiae*, collected over 18 time points throughout the cell cycle[80]. Colors indicate cluster membership based on a *k*-means clustering ($k = 4$). (**a**) Scatter plot showing a projection of the profiles on the first two principal components obtained by PCA. (**b**) Profile plot of gene expression across all 18 time points, including *k*-means cluster information. Genes in the red and blue clusters appear active in the G1 and S phase of the cell cycle, respectively. Phase assignments for yellow and green clusters are unclear. (**c**) Heat map of the profiles. Colors represent abundance (red, higher than control; blue, lower than control; white, no change). Rows of the heat map have been reordered according to a hierarchical clustering, represented by the dendrogram. The color bars between the dendrogram and heat map indicate the *k*-means clusters, allowing comparison of the two clustering results. Images made with R (http://www.r-project.org/).

Profile plots. Profile plots (**Fig. 3b**), also known as parallel coordinate plots[135], visualize the expression levels of a large number of transcripts across all samples. Thus, they provide insight into the patterns of correlation between samples and expression levels. For instance, at a glance, the scientist can determine whether a transcript is expressed constitutively in all conditions or whether it is only expressed in a single condition, such as a particular tissue or phase of the cell cycle. Furthermore, it is possible to generate hypotheses about trends, such as increasing expression levels for a transcript over time after a stimulus, or differential expression of a transcript—for instance, between samples of diseased and normal tissue. Because many profiles are shown in the same plot, the scientist can interpret such observations in the context of the overall data set.

A profile plot can also be queried visually for transcripts with a particular behavior, such as low expression in one set of samples and high in another set, or for profiles that are similar to that of a transcript of interest. A substantial disadvantage of profile plots is that, owing to the manner in which they are constructed, profiles overlap, severely limiting the number of profiles that can be visualized effectively at the same time.

Heat maps. Heat maps[136,137] (**Fig. 3c**) are the most commonly used visualization method for expression matrices[138] and can be generated using most tools. Like profile plots, heat maps visualize the abundance of each transcript in each sample, but the profiles do not overlap, which means that more profiles can be visualized effectively. However, the size of the heat map grows with the number of profiles, so that the available screen space is often a limiting factor.

A key aspect of heat map visualization is the reordering of the rows, which ensures that similar profiles are placed near each other. Typically this reordering is done using hierarchical clustering[137], and a dendrogram showing the hierarchy is usually arranged immediately adjacent to the heat map (**Fig. 3**). This combined view helps a scientist to see groups of genes that have a similar expression pattern. The dendrogram conveys which genes are clustered together, and also which genes are outliers with an unusual expression pattern. The heat map allows the scientist to see in more detail which features of the expression pattern are shared by gene clusters. For example, genes in a cluster may have a peak expression at about the same time in an experiment.

partners are, or whether they are located in the same subcellular compartment. In contrast, when expression profiles are shown only in the nodes of the network, this type of analysis is not possible because coexpressed genes are not necessarily located next to each other in the visualization. However, there is a trade-off between the flexibility provided by linked views and the convenience of being able to see expression profiles and interactions without having to consult two separate visualizations.

Network clustering and correlation networks. Recently, there has been increased interest in a new kind of clustering method—called 'network clustering'[37]—that is less susceptible to noise and can lead to more accurate identification of functionally related genes than established clustering methods (**Box 2**). Network clustering of gene expression data is done using so-called 'correlation networks', in which each gene is a node and each edge indicates coexpression of two genes under the conditions of the experiment[43]. As well as being an improved way to calculate clusters, correlation networks allow the scientist to interactively explore gene expression data sets using many of the rich set of network visualization tools that have been developed for visualizing protein interaction networks. The use of correlation networks for gene expression data is as yet supported by relatively few tools, including BioLayout Express 3D[37]—which has been developed specifically for this purpose—and Cytoscape, using either the MCODE[18] or ClusterMaker plug-in (http://www.cgl.ucsf.edu/cytoscape/). However, we anticipate that correlation networks may become one of the established methods for interpreting gene expression data sets.

Metabolic profile data

A wide variety of spectroscopic methods are being used for high-throughput studies of small-molecule metabolites[44], two of the most popular being mass spectrometry and nuclear magnetic resonance spectroscopy (**Box 1** and **Supplementary Table 1**). Typically, present methods identify hundreds of metabolites per experiment. Additionally, many as-yet-unidentified compounds can be reproducibly detected. These experimental data are collected in several public repositories, the largest of which is now SetupX[45], containing ~20,000 samples from more than 300 studies.

The general goal in analyzing metabolite profiles is to gain detailed insight into the molecular mechanisms of cellular metabolic pathways. The identification of molecules that may be used as reliable biomarkers of disease is also of great interest. Metabolite profiles are typically analyzed to find sets of metabolites with similar profiles or to measure the impact of genetic modifications, drugs and other biotic or abiotic factors on the metabolome of an organism.

As with gene and protein profiles, visualization is key in these analyses, and the same, or very similar, methods (**Box 2**) and tools (**Table 3**) are typically used. As for gene expression data, one of the key visualization methods in metabolomics involves the enrichment of metabolic pathways with visualizations of metabolite concentrations (**Fig. 4a**), and often the same visual representations as for gene expression data can be used (**Fig. 2b–g**). Visualizing such enriched metabolic pathways can be very useful in understanding the concerted changes of metabolite pools within the cell. In addition, enriched pathways can help to identify metabolites that should be present according to enzymatic reactions contained in the metabolic pathway but that have not been detected in the measurements. If such metabolites are identified, further experiments attempting to detect these molecules can be conducted. Many visualization tools (for example, MapMan[46], Pathway Tools Omics Viewer[47], KaPPA-view[48], PathwayExplorer[49] and ProMeTra[41]) have been developed to facilitate enriched views of metabolic pathways, usually with close integration of metabolic pathway databases. These tools overlay metabolite profiles primarily on static images of pathways obtained from sources such as KEGG[29] or MetaCyc[50]-based databases.

Pathway editing

The analysis of new experimental data sets, as outlined above, usually produces new insight into biological processes, which may be used to modify existing pathways or to create new pathways. A wide range of tools is available that support pathway editing (**Table 2**); the choice of which tool to use depends on the specific requirements of the task at hand.

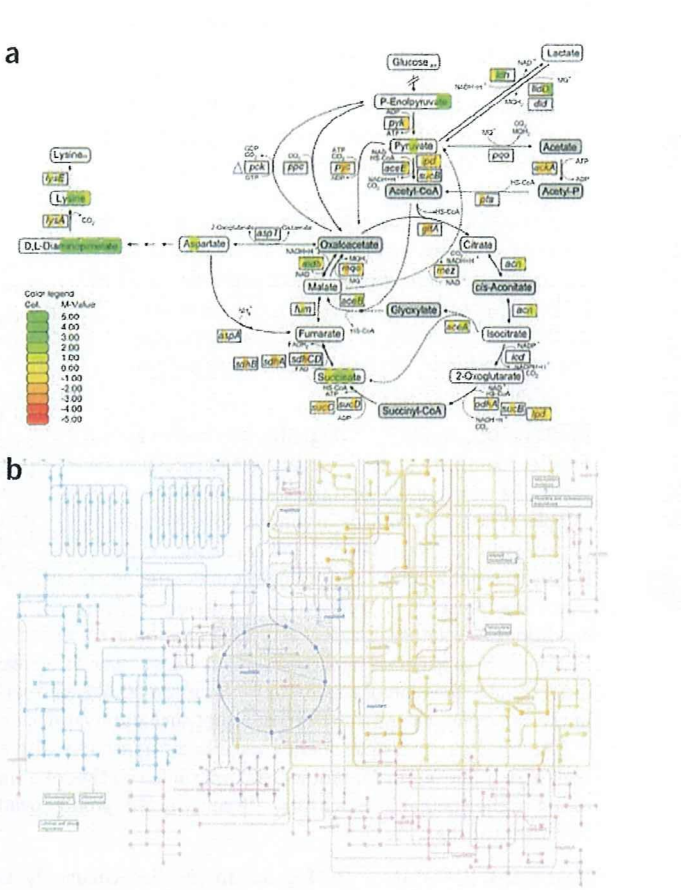

Figure 4 | Visualization of metabolic pathways and profile data. (**a**) A part of the glycolysis and citric acid cycle pathway in *Corynebacterium glutamicum* DM1730 overlaid with changes in metabolite concentrations and gene expression across five time points in relation to a reference time point. The visualization was created in ProMeTra[41] (as in **Fig. 2b**). Nodes shaded gray indicate metabolites for which no concentration data were available. (**b**) Enlarged iPath/KEGG Atlas image showing the glycolysis pathway in the context of other parts of the metabolic system. Yellow, amino acid metabolism, purple, energy metabolism. The shaded area corresponds to the citric acid cycle shown in **a**.

For building pathways from scratch or editing existing pathways, tools such as GenMAPP 2, PathVisio, and VANTED are useful, as they are designed for to assist the manual task of arranging nodes and edges. To this category also belong Cell Designer[51] and JDesigner[52], which further support pathway simulations by means of kinetic modeling. The insights gained from these simulations often lead to new hypotheses, which can then be tested in further experiments.

Manual layout of pathways quickly becomes tedious as the size of the pathway grows. Fortunately, a range of automated layout methods have been developed, each addressing specific layout requirements. Typically, these methods will arrange the network to highlight the overall state changes that occur—for example, making sure that all interactions point from left to right, and thus creating an overall causal flow from left to right. Automated layout can be particularly useful for updating large networks when new knowledge (nodes or interaction edges) becomes available. For example, PATIKA[53] has an automated layout method that shows the causal flow of events through different subcellular compartments. This is particularly useful for depicting signaling networks[27]. Although these specialized automated layout methods are useful, they are usually of low quality compared to manually laid out pathways created by human experts and often require manual editing in addition; however, judging by recent progress, we expect these method to continue to improve and to become increasingly useful[54].

For very large pathways, it can be important to use compact visual representations and pathway layouts that reduce the amount of detail shown. A very clear illustration of such a concise visual representation is iPath, which combines 120 KEGG pathways into a single, vast pathway map that provides an overview of all metabolism in an organism (**Fig. 4b**). Scientists can zoom into parts of the map to navigate to individual pathways.

Future perspectives

Systems biology is still rapidly evolving, which can make it difficult for tool developers to know which visualization tasks are the most important ones. However, as the field matures, the key tasks will likely become clearer, and the requirements and limitations of current visualization methods will become better understood[5]. This process will also be aided by insights from the emerging field of visual analytics[55], which specifically studies the role of visualization in the larger process of understanding and interpreting data. Visual analytics methods have begun to be applied to studying the connection between visualization and analytical reasoning in systems biology[5,56].

We anticipate that the near future will bring significant improvements in automated pathway and network layout to better match biologists' needs[54,57]. Innovation will continue to give more and better choices for the representations of nodes, edges and overlay information, as well as better ways to convey dynamic properties and to compare networks. Crucially, we expect that usability will improve, partly through improved navigation methods that help users manage large and complex networks[23,25,58].

Today, many tools for network and pathway visualization are stand-alone applications (**Tables 1** and **2**); however, there is a trend toward web-based applications, often coupled tightly to underlying databases. Web-based tools show great promise for facilitating collaboration between scientists at different locations[59–61], and several projects have recently been launched that are aimed at community-based collaborative editing of biological network data—notably Payaologue (http://celldesigner.org/payao/) and WikiPathways[62].

As experimental methods enable scientists to tackle larger and more complex systems, it is likely that significant innovations will be needed for visualizing future data sets. One possible direction for future network visualization tools would be to move beyond the standard two-dimensional layout, and tool developers are already exploring three-dimensional layouts (for example, BioLayout Express 3D[37]), combinations of both three-dimensional layouts and time (for example, E-Cell 3D, http://tinyurl.com/ecell3d/), or layouts that mix aspects of two and three dimensions (for example, Arena3D[63]). In addition, systems biologists may well be among the early adopters of innovations in hardware, such as multi-touch interfaces and larger, high-resolution displays[64].

As systems biology has evolved very quickly over the last decade, some of the difficulties faced by end-users today arise not from the intrinsic complexity of data but from a lack of standards. Biological pathways and networks are now distributed in over 300 web resources[4]—and in a field as interdisciplinary as systems biology, there is an obvious strength in such diversity. However, the field would clearly benefit from a parallel effort toward a consolidated resource, and we would like to add our voices to a call for a consolidated database, similar to the worldwide Protein Data Bank for three-dimensional structures[65].

The situation is somewhat better with file formats used to store interaction data, pathways and biochemical models. Although many formats are used, several have emerged as *de facto* standards for the exchange of pathway and network data—for example, PSI-MI[66] for protein interaction data, BioPAX (http://www.biopax.org/) for pathways and interaction networks, Systems Biology Markup Language (SBML)[67] for models of biochemical reactions and gene regulation and CellML[68] for exchange of a range of different biological models. In regard to graphical notation, there has recently been a significant community-driven proposal (Systems Biology Graphical Notation, SBGN[69]) toward developing a more unified standard, and several tools already support the creation and visualization of networks using this standard (see **Table 2**).

Ultimately, systems biology seeks to provide insights into the processes of organelles, cells, organs and even whole organisms. Fulfilling this ambitious goal requires still further development in visualization methods; in particular, better integration with visualization of other kinds of data, such as imaging data[70], macromolecular structures[71], genomes[72], and phylogenies[73]. Efforts to build such integrated visualization platforms have begun (for example, Visible Cell[74]), and in fact, many tools that bridge different data types and disciplines are already in place; for instance, there are tools that map transcript abundance (or, if available, protein abundance) onto chromosomal location[75] and onto three-dimensional anatomical representations of tissue[76,77]. However, truly integrated visualization of systems biology data across the entire range of possible data types is still very much in its infancy.

Note: Supplementary information is available on the Nature Methods website.