

Systems Biology Graphical Notation: Activity Flow Diagram Level 1

Draft

Date: August 14, 2009

Disclaimer: This is a working draft of the SBGN Activity Flow Diagram Level 1 specification. It is not a normative document.

Editors:

Huaiyu Mi
Falk Schreiber
Nicolas Le Novère
Stuart Moodie
Anatoly Sorokin

*SRI International, USA
IPK Gatersleben & MLU Halle, Germany
EMBL European Bioinformatics Institute, UK
CSBE, University of Edinburgh, UK
University of Edinburgh, UK[7pt]*

Principal Authors:

To discuss any aspect of SBGN, please send your messages to the mailing list sbgn-discuss@sbgn.org. To get subscribed to the mailing list or to contact us directly, please write to sbgn-team@sbgn.org. Bug reports and specific comments about the specification should be entered in the issue tracker http://sourceforge.net/tracker/?group_id=178553&atid=1082245.



Preface

Acknowledgements

The authors are grateful to all the attendees of the SBGN meetings, as well as to the subscribers of the sbgn-discuss@sbgn.org mailing list.

The development of SBGN was mainly supported by a grant from the Japanese *New Energy and Industrial Technology Development Organization* (NEDO, <http://www.nedo.go.jp/>). The *Okinawa Institute of Science and Technology* (OIST, <http://www.oist.jp/>), the *AIST Computational Biology Research Center* (AIST CBRC, <http://www.cbrc.jp/index.eng.html>) the *British Biotechnology and Biological Sciences Research Council* (BBSRC, <http://www.bbsrc.ac.uk/>) through a Japan Partnering Award, the *European Media Laboratory* (EML Research gGmbH, <http://www.eml-r.org/>), and the *Beckman Institute at the California Institute of Technology* (<http://bnmc.caltech.edu>) provided additional support for SBGN workshops. Some help was provided by the *Japan Science and Technology Agency* (JST, <http://www.jst.go.jp/>) and the *Genome Network Project* of the Japanese Ministry of Education, Sports, Culture, Science, and Technology (MEXT, <http://www.mext.go.jp/>) for the development of the gene regulation network aspect of SBGN, and from the *Engineering and Physical Sciences Research Council* (EPSRC, <http://www.epsrc.ac.uk>) during the redaction of the specification.

Notes on typographical conventions

The concept represented by a glyph is written using a normal font, while a *glyph* means the SBGN visual representation of the concept.

Contents

Preface	ii	2.7.2 Glyph: <i>Negative influence</i>	17
1 What is the Systems Biology Graphical Notation?	1	2.7.3 Glyph: <i>Unknown influence</i>	18
1.1 History of SBGN development	1	2.7.4 Glyph: <i>Necessary stimulation</i>	18
1.2 The three languages of SBGN	2	2.8 Logical operators	19
1.3 SBGN levels	3	2.8.1 Glyph: <i>And</i>	19
1.4 Developments, discussions, and notifications of updates	4	2.8.2 Glyph: <i>Or</i>	19
		2.8.3 Glyph: <i>Not</i>	20
2 Activity Flow Diagram glyphs	5	3 Activity Flow Diagram grammar	22
2.1 Overview	5	3.1 Overview	22
2.2 Controlled vocabularies used in SBGN Activity Flow Level 1	6	3.2 Concepts	22
2.2.1 Activity node material types	6	3.3 The conceptual model	22
2.2.2 Activity node conceptual types	7	3.4 Syntax	22
2.2.3 Physical characteristics of compartments	7	3.5 Semantic rules	22
2.2.4 Cardinality	7	3.5.1 Namespaces	22
2.3 Activity nodes	8	3.5.2 Compartment spanning	22
2.3.1 Glyph: <i>Biological activity</i>	8	3.5.3 Compartments	23
2.3.2 Glyph: <i>Perturbation</i>	9	3.5.4 Modulation	23
2.3.3 Glyph: <i>Phenotype</i>	10	3.5.5 Submaps	23
2.4 Auxiliary units	10	4 Layout Guidelines for a Activity Flow Diagram	25
2.4.1 Glyph: <i>Unit of information</i>	10	4.1 Introduction	25
2.5 Container nodes	12	4.2 Layout guidelines	26
2.5.1 Glyph: <i>Compartment</i>	12	4.2.1 Requirements	26
2.6 Glyph: <i>Submap</i>	14	4.2.2 Recommendations	27
2.7 Modulation arcs	16	4.2.3 Additional suggestions	27
2.7.1 Glyph: <i>Positive influence</i>	16	A Examples	29
		B Reference card	33

Chapter 1

What is the Systems Biology Graphical Notation?

The goal of the **S**ystems **B**iology **G**raphical **N**otation (SBGN) is to standardize the graphical/visual representation of essential biochemical and cellular processes studied in systems biology. SBGN defines a comprehensive set of symbols with precise semantics, together with detailed syntactic rules defining their use. It also describes the manner in which such graphical information should be interpreted.

Standardizing graphical notations for describing biological interactions is an important step towards the efficient and accurate transmission of biological knowledge between different communities. Traditionally, diagrams representing interactions among genes and molecules have been drawn in an informal manner, using simple unconstrained shapes and edges such as arrows. Until the development of SBGN, no standard agreed-upon convention existed defining exactly how to draw such diagrams in a way that helps readers interpret them consistently, correctly, and unambiguously. By standardizing the visual notation, SBGN can serve as a bridge between different communities such as computational and experimental biologists, and even more broadly in education, publishing, and more.

For SBGN to be successful, it must satisfy a majority of technical and practical needs, and must be embraced by the community of researchers in biology. With regards to the technical and practical aspects, a successful visual language must meet at least the following goals:

1. Allow the representation of diverse biological objects and interactions;
2. Be semantically and visually unambiguous;
3. Allow implementation in software that can aid the drawing and verification of diagrams;
4. Have semantics that are sufficiently well defined that software tools can convert graphical models into formal models, suitable for analysis if not for simulation;
5. Be unrestricted in use and distribution, so that the entire community can freely use the notation without encumbrance or fear of intellectual property infractions.

This document defines the *Activity Flow* visual language of SBGN. As explained more fully in Section 1.2, Activity Flow diagrams are one of three views of a model offered by SBGN. It is the product of many hours of discussion and development by many individuals and groups. In the following sections, we describe the background, motivations, and context of Activity Flow diagrams.

1.1 History of SBGN development

Although problems surrounding the representation of biological pathways has been discussed for a long time, see for instance [1], the effort to create a well-defined visual notation was pioneered

by Kurt Kohn with his Molecular Interaction Map (MIM), a notation defining symbols and syntax to describe the interactions of molecules [2]. MIM is essentially a variation of the entity-relationship diagrams [3]. Kohn's work was followed by numerous other attempts to define both alternative notations for diagramming cellular processes (e.g., the work of Pirson and colleagues [4], BioD [5], Patika [6, 7], and others), as well as extensions of Kohn's notation (e.g., the Diagrammatic Cell Language of Maimon and Browning [8]).

Kitano originated the idea of having multiple views of the *same* model. This addresses two problems: no single view can satisfy the needs of all users, and a given view can only represent a subset of the semantics necessary to express biological knowledge. Kitano proposed the development of process diagrams, entity-relationship diagrams, timing charts (to describe temporal changes in a system), and abstract flow charts [9]. The Process Diagram notation was the first to be fully defined using a well-delineated set of symbols and syntax [10]. It led to a desire to establish a unified standard for graphical representation of biochemical entities, and from this arose the current SBGN effort. Separately and roughly concurrently, other groups designed similar notations, for example the Edinburgh Pathway Notation [11] or Patika [6, 7]. All of these efforts began to attract attention as more emphasis in biological research was placed on networks of interactions and not just characterization of individual entities.

In 2005, thanks to funding from the Japanese agency *The New Energy and Industrial Technology Development Organization* (NEDO, <http://www.nedo.go.jp/>), Kitano initiated the Systems Biology Graphical Notation (SBGN) project as a community effort. The first SBGN workshop was held in February 2006 in Tokyo, with over 30 participants from major organizations interested in this effort. From the in-depth discussions held during that meeting emerged a set of decisions that are the basis of the current SBGN specification. These decisions are:

- SBGN should be made up of two different visual grammars, describing Entity Relationship and Process Diagram diagrams (called *State Transition* diagrams at the time). See Section 1.2.
- In order to promote wide acceptance, the initial version(s) of SBGN should stick to at most a few dozens symbols that non-specialists could easily learn.

The second SBGN workshop was held in October, 2006, in Yokohama, Japan. This meeting featured the first technical discussions about which symbols to include in SBGN Level 1, as well as discussions about the syntax, semantics, and layout of graphs. A follow-up technical meeting was held in March, 2007, in Heidelberg, Germany; the participants of that meeting fleshed out most of the design of SBGN. The third SBGN workshop, held in Long Beach in October, 2007, was dedicated to reaching agreement on the final outstanding issues of notation and syntax. The participants of that meeting collectively realized that a third language would be necessary: the Activity Flow diagrams. The specification for the Process Diagram language was finalized and largely completed during a follow-up technical meeting held in Okinawa, Japan, in January, 2008. At this meeting, attendees also held the first in-depth discussions about the syntax of the Entity Relationship language. The specification for the Activity Flow language was initially discussed during a meeting held in Rostock, Germany, in October, 2008. SBGN workshops are an opportunity for public discussions about SBGN, allowing interested persons to learn more about SBGN and help identify needs and issues. More meetings are expected to be held in the future, long after this specification document has been issued.

1.2 The three languages of SBGN

Readers may well wonder, why are there *three* languages in SBGN? The reason is that this approach solves a problem that was found insurmountable any other way: attempting to include all relevant facets of a biological system in a single diagram causes the diagram to become hopelessly complicated and incomprehensible to human readers.

The three different notations in SBGN correspond to three different *views* of the same model. These views are representations of different classes of information, as follows:

1. *Process Diagram*: the causal sequences of molecular processes and their results
2. *Entity Relationship*: the interactions between entities irrespective of sequence
3. *Activity Flow*: the flux of information going from one entity to another

In the Process Diagram view, each node in the diagram represents a given *state* of a species, and therefore a given species may appear multiple times in the same diagram if it represents the same entity in different states. Conversely, in the Entity Relationship view, a given species appears only once in a diagram. Process Diagrams are suitable for following the temporal aspects of interactions, and are easy to understand. The drawback of the Process Diagram, however, is that because the same entity appears multiple times in one diagram, it is difficult to understand which interactions actually exist for the entity. Conversely, Entity Relationship diagrams are suitable for understanding relationships involving each molecule, but the temporal course of events is difficult or impossible to follow because Entity Relationship diagrams do not describe the sequence of events.

Process Diagrams can quickly become very complex. Moreover, when diagramming a biochemical network, one often wants to ignore the biochemical basis underlying the action of one entity on the activity of another. A common desire is to represent only the flow of activity between nodes, without representing the transitions in the states of the nodes. This is the motivation for the creation of the Activity Flow view. Activity Flow diagrams permit the use of *modulation*, *stimulation* and *inhibition* and allow them to point to State/Entity nodes rather than process nodes. The Activity Flow view is thus a hybrid between Process Diagram and Entity Relationship diagrams. It is particularly convenient for representing the effect of perturbations, whether genetic or environmental in nature.

A recurring argument in SBGN development is that these three types of diagrams should be merged into one. Unfortunately, each view has such different meanings that merging them would compromise the robustness of the representation and destroy the mathematical integrity of the notation system. While having three different notations makes the overall system more complex, much of the complexity and increase in burden on learning is mitigated by reusing most of the same symbols in all three notations. It is primarily the syntax and semantics that change between the different views, reflecting fundamental differences in the underlying mathematics of what is being described.

1.3 SBGN levels

It was clear at the outset of SBGN development that it would be impossible to design a perfect and complete notation right from the beginning. Apart from the prescience this would require (which, sadly, none of the authors possess), it also would likely require a vast language that most newcomers would shun as being too complex. Thus, the SBGN community followed an idea used in the development of the Systems Biology Markup Language (SBML; [12]): stratify language development into levels.

A *level* of SBGN represents a set of features deemed to fit together cohesively, constituting a usable set of functionality that the user community agrees is sufficient for a reasonable set of tasks and goals. Capabilities and features that cannot be agreed upon and are judged insufficiently critical to require inclusion in a given level, are postponed to a higher level. In this way, SBGN development is envisioned to proceed in stages, with each higher SBGN level adding richness compared to the levels below it.

1.4 Developments, discussions, and notifications of updates

The SBGN website (<http://sbgn.org>) is a portal for all things related to SBGN. It provides a web forum interface to the SBGN discussion list (sbgn-discuss@sbgn.org) and information about how anyone may subscribe to it. The easiest and best way to get involved in SBGN discussions is to join the mailing list and participate.

Face-to-face meetings of the SBGN community are announced on the website as well as the mailing list. Although no set schedule currently exists for workshops and other meetings, we envision holding at least one public workshop per year. As with other similar efforts, the workshops are likely to be held as satellite workshops of larger conferences, enabling attendees to use their international travel time and money more efficiently.

Notifications of updates to the SBGN specification are also broadcast on the mailing list and announced on the SBGN website.

Chapter 2

Activity Flow Diagram glyphs

This chapter provides a catalog of the graphical symbols available for representing entities in Activity Flow diagrams. In Chapter 3 beginning on page 22, we describe the rules for combining these glyphs into a legal SBGN Activity Flow, and in Chapter 4 beginning on page 25, we describe requirements and guidelines for the way that diagrams are visually organized.

2.1 Overview

To set the stage for what follows in this chapter, we first give a brief overview of some of the concepts in the Activity Flow notation with the help of an example shown in Figure 2.1 on the next page.

The diagram illustrates the regulation of peroxisome proliferator-activated receptor delta (PPAR delta, a nuclear hormone receptor) on brown fat metabolism, a redraw from Fig 7E of Pan et. al [13]. The rectangle nodes represent *biological activities* - activities from biological materials. The type of material is indicated in the units of information decorated on the activity nodes (See Section 2.3.1). Each biological activity can influence, or be influenced by, other biological activities, and such relationships are represented in Activity Flow by lines with arrows and other decorations. It should be noted that the essence of Activity Flow is to show the flow of activities from one entity to another or within the same entity. For example, in the diagram, it shows that PPAR δ positively influences the Twist gene expression. The underlying mechanisms of how the influence occurs may not be known and is not captured in the diagram. If the mechanism is known, the details should be described in annotation or captured in other SBGN languages, such as Process Diagram and/or Entity Relationship.

Table 2.1 summarizes the different SBGN abstractions described in this chapter.

Component	Abbrev.	Role	Examples
Activity node	AN	A functional unit that can affect, or be affected by, another functional unit.	Biological activity
Container node	CN	An encapsulation of one or more other SBGN constructs	Complexes, compartments
Modulation arc	MA	Links between different activities to indicate influences.	Positive influence, Negative influence
Logical operators	—	Combines one or several inputs into one output	Boolean <i>and</i> , <i>or</i> , <i>not</i>

Table 2.1: Summary of Activity Flow components and their roles.

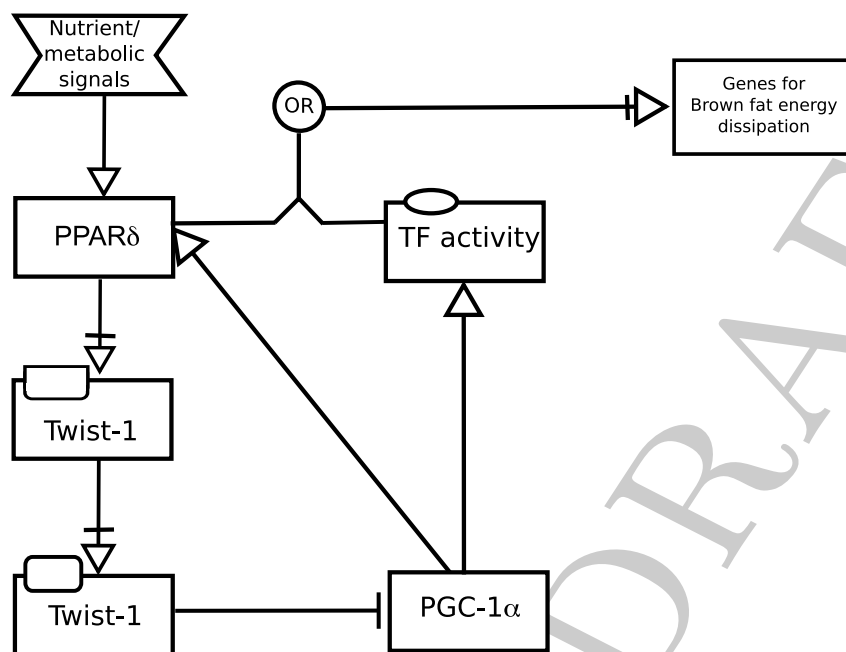


Figure 2.1: This example of Activity Flow depicts the effect of nutrients and metabolic signals on brown fat metabolism through PPAR δ . The signal, shown as a perturbation, positively influences the nuclear hormone receptor PPAR δ , which in turn stimulates the Twist 1 gene expression. Please note the different units of information on Twist-1 activity nodes that indicate the activity from different biological materials (gene and macromolecule). The Twist-1 protein negatively influence the PGC-1 α activity. PGC-1 α positively influences PPAR δ and other unspecified transcription factor activity to stimulate the expression of genes for brown fat energy dissipation. Therefore, the Twist-1, induced by PPAR δ , serves as a negative feedback regulator of PGC-1 α in brown fat metabolism.

2.2 Controlled vocabularies used in SBGN Activity Flow Level 1

What controlled vocabulary should we use to describe the activity? SBO? GO Molecular function?

Some glyphs in SBGN Activity Flow can contain particular kinds of textual annotations conveying information relevant to the purpose of the glyph. These annotations are *units of information* (Section 2.4.1). An example is in the case of multimers, which can have a unit of information conveying the number of monomers composing the multimer. Other cases are described throughout the rest of this chapter.

2.2.1 Activity node material types

The material type of an AFN indicates its chemical structure. A list of common material types is shown in Table 2.2 on the next page, but others are possible. The values are to be taken from the Systems Biology Ontology (<http://www.ebi.ac.uk/sbo/>), specifically from the branch having identifier **SBO:0000240** (*material entity under participant*→*physical participant*). The labels are defined by SBGN Activity Flow Level 1.

The material types are in contrast to the *conceptual types* (see below). The distinction is that material types are about physical composition, while conceptual types are about roles. For example, a strand of RNA is a physical artifact, but its use as messenger RNA is a role.

Name	Label	SBO term
Non-macromolecular ion	mt:ion	SB0:0000327
Non-macromolecular radical	mt:rad	SB0:0000328
Ribonucleic acid	mt:rna	SB0:0000250
Deoxribonucleic acid	mt:dna	SB0:0000251
Protein	mt:prot	SB0:0000297
Polysaccharide	mt:psac	SB0:0000249

Table 2.2: A sample of values from the material types controlled vocabulary (Section 2.2.1).

2.2.2 Activity node conceptual types

An AFN's *conceptual type* indicates its function within the context of a given Activity Flow. A list of common conceptual types is shown in Table 2.3, but others are possible. The values are to be taken from the Systems Biology Ontology (<http://www.ebi.ac.uk/sbo/>), specifically from the branch having identifier **SB0:0000241** (*conceptual entity* under *participant*→*physical participant*). The labels are defined by SBGN Activity Flow Level 1.

Name	Label	SBO term
Gene	ct:gene	SB0:0000243
Transcription start site	ct:tss	SB0:0000329
Gene coding region	ct:coding	SB0:0000335
Gene regulatory region	ct:grr	SB0:0000369
Messenger RNA	ct:mRNA	SB0:0000278

Table 2.3: A sample of values from the conceptual types vocabulary (Section 2.2.2).

2.2.3 Physical characteristics of compartments

SBGN Activity Flow Level 1 defines a special unit of information for describing certain common physical characteristics of compartments. Table 2.4 lists the particular values defined by SBGN Activity Flow Level 1. The values correspond to the Systems Biology Ontology branch with identifier **SB0:0000255** (*physical characteristic* under *quantitative parameter*).

Name	Label	SBO term
Temperature	pc:T	SB0:0000147
Voltage	pc:V	SB0:0000259
pH	pc:pH	SB0:0000304

Table 2.4: A sample of values from the physical characteristics vocabulary (Section 2.2.3).

2.2.4 Cardinality

SBGN Activity Flow Level 1 defines a special unit of information usable on multimers for describing the number of monomers composing the multimer. Table 2.5 on the following page shows the way in which the values must be written. Note that the value is a unitary number, and not (for example) a range. There is no provision in SBGN Activity Flow Level 1 for specifying a range in this context because it leads to problems of entity identifiability.

Name	Label	SBO term
cardinality	N:#	SBO:0000364

Table 2.5: The format of the possible values for the cardinality unit of information (Section 2.2.4). Here, # stands for the number; for example, “N:5”.

2.3 Activity nodes

An activity node (AN) represents the activity of an entity or an entity pool, but not the entities themselves. For instance, multiple activity nodes can be used to represent different activities of a particular entity, while one activity node can be used to represent the activity of a complex multimer. In addition to activities of material entities, SBGN Activity Flow Level 1 represents activity from two conceptual entities: *perturbation*, *observable*. Auxiliary units such as state variables are not shown on the activity nodes. Each activity is displayed only once in one compartment.

2.3.1 Glyph: *Biological activity*

SBGN Activity Flow Level 1 uses one glyph to represent activities from all biological entities, collectively they are called *biological activity*. The nature of the molecule that the activity comes from, eg., simple chemical or macromolecule, can be encoded in the *units of information* (Section 2.4.1).

It should be noted that the *biological activity* is not equivalent to a biological entity per se. A biological activity can come from one biological entity, a part of an entity, or a combination of them. It is up to the users to determine how to represent it in their diagram. For example, a protein kinase receptor such as an EGF receptor, has two activities, the binding activity that allows the extracellular part of the receptor to bind to the ligand, and the kinase activity that is capable of phosphorylating the downstream protein and initiating the intracellular signaling. The user can choose to use two nodes to represent each activity, or to use one node to represent the overall “EGF receptor activity”.

SBO Term:

SBO:0000412 ! biological activity

Container:

An *biological activity* is represented by a rectangle, as shown in Figure 2.2 on the next page.

Label:

An *biological activity* is identified by a label placed in an unbordered box containing a string of characters. The characters can be distributed on several lines to improve readability, although this is not mandatory. The label box must be attached to the center of the container. The label may spill outside of the container.

Auxiliary items:

A *biological activity* can carry a *unit of information* (Section 2.4.1), which can provide information such as the nature of the entity from which the activity originated. Specific glyphs are used to represent different types of entities (Section 2.4.1). The center of the bounding box of a *unit of information* is located on the mid-line of the border of the macromolecule. The label in the *unit of information*, which is optional, indicates the name of the molecule where the activity comes from, as shown in Figure 2.3 on the following page



Figure 2.2: The Activity Flow glyph for biological activity.

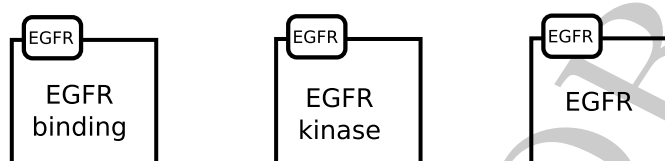


Figure 2.3: An example of Activity Flow glyphs of EGFR activities. Since EGFR protein has both binding and kinase activities, each of those activity can be represented by different nodes, labeled as EGFR binding and EGFR kinase. One node can be used to represent the overall activity of EGFR. The label in the unit of information indicates the protein that the activities come from. In this example, all three activities come from the same EGFR protein

2.3.2 Glyph: *Perturbation*

Biochemical networks can be affected by external influences. Those influences can be well-defined physical perturbations, such as a light pulse or a change in temperature; they can also be more complex and not well-defined phenomena, for instance, glucose deprivation, stress. For these situations, SBGN provides the perturbation glyph. In SBGN Activity Flow Level 1, a *perturbation* is usually connected as an origin to the *modulation arcs* or *logic operators*.

SBO Term:

SBO:0000357 ! perturbation

Container:

A *perturbation* is represented by a modified hexagon having two opposite concave faces, as illustrated in Figure 2.4.

Label:

A *perturbation* is identified by a label placed in an unbordered box containing a string of characters. The characters can be distributed on several lines to improve readability, although this is not mandatory. The label box must be attached to the center of the *perturbation* container. The label may spill outside of the container.



Figure 2.4: The Activity Flow glyph for perturbation.

2.3.3 Glyph: *Phenotype*

A biochemical network can generate phenotypes or affect biological processes. Such processes can take place at different levels and are independent of the biochemical network itself. To represent these processes in a diagram, SBGN defines the *phenotype* glyph.

SBO Term:

SBO:0000358 ! phenotype

Container:

A *phenotype* is represented by an elongated hexagon, as illustrated in Figure 2.5.

Label:

An *phenotype* is identified by a label placed in an unbordered box containing a string of characters. The characters can be distributed on several lines to improve readability, although this is not mandatory. The label box must be attached to the center of the *phenotype* container. The label may spill outside of the container.



Figure 2.5: *The Activity Flow glyph for phenotype.*

2.4 Auxiliary units

2.4.1 Glyph: *Unit of information*

When representing biological activities, it is often useful to illustrate the nature of the entity where the activity is originated, eg., whether the activity is from a macromolecule (protein or nucleic acid), or from a chemical compound. The SBGN Activity Flow Level 1 *unit of information* is used in this situation to add such information to a glyph. It represents the information in two ways. First, different symbols are used to represent the nature of the entity where the activity is from. These symbols are identical to the *entity pool node* symbols in SBGN Process Diagram. Second, names of the entity (gene names, protein names) are usually provided as labels in the *unit of information* container.

SBO Term:

Not applicable.

Container:

A unit of information is represented by containers of different shapes, depending on the nature of the entity where the biological activity is from. There are a total of five types of unit of information, as shown in Figure 2.6 on the next page. Below is a summary of the five glyphs.

A. macromolecule – A unit of information of a macromolecule is represented by a rectangle with rounded corners, as illustrated in (A) of Figure 2.6 on the following page. This container is used to decorate a biological activity that is originated from a macromolecule, such as a protein, a nucleic acid, or a complex sugar.

B. genetic – A unit of information of a genetic entity is represented by a rectangle whose bottom half has rounded corners, as shown in (B) of Figure 2.6 on the next page.

C. simple chemical – A unit of information of a simple chemical is represented by a circular container, as shown in (C) of Figure 2.6 on the following page.

D. unspecified entity – A unit of information of an unspecified entity is represented by an elliptic container, as shown in (D) of Figure 2.6. It is used to decorate a biological activity that is originated from an unspecified entity.

E. complex – A unit of information of a complex is represented by an octagon as shown in E of Figure 2.6. It is used to decorate a biological activity that is originated from a complex.

The long side of the glyphs above (except for simple chemical) should be oriented parallel to the border of the *AFN* being annotated by the *unit of information*. The center of the bounding box of a *state of information* should be located on the mid-line of the border of the *AFN*.

Label:

A *unit of information* is identified by a label placed in an unbordered box containing a string of characters. The characters can be distributed on several lines to improve readability, although this is not mandatory. The label box must be attached to the center of the container. The label may spill outside of the container. The label defines the information carried by the *unit of information*.

Auxiliary items:

A *unit of information* does not carry any auxiliary items.

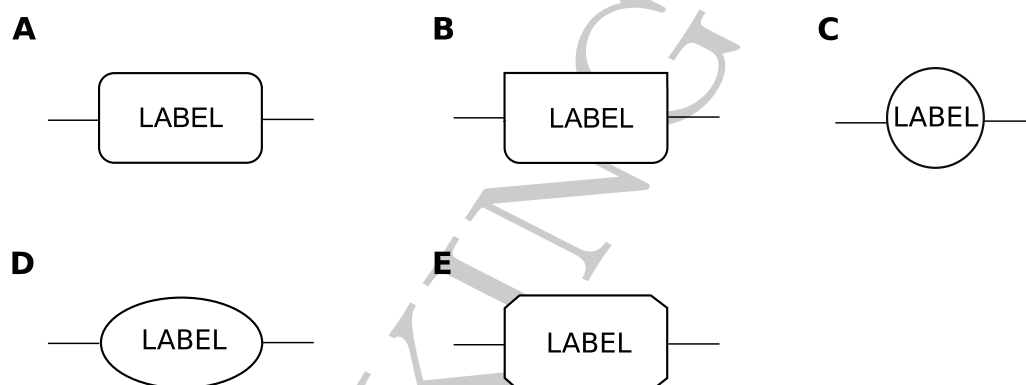


Figure 2.6: The Activity Flow glyph for unit of information.

Figure 2.7 shows examples units of information used on biological activity nodes to illustrate the properties of the entities that the activities are originated from.

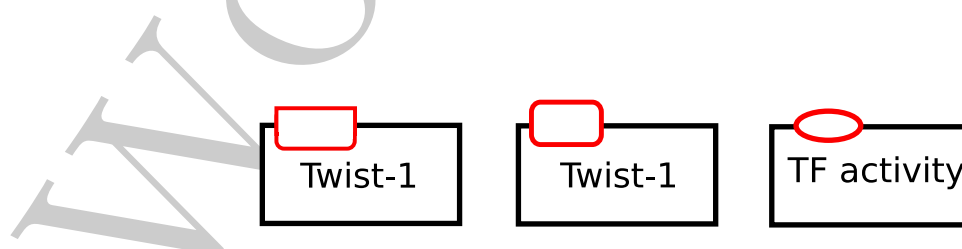


Figure 2.7: Examples of unit of information used on biological activity node to indicate that the activity is from a gene, a macromolecule, or unspecified.

2.5 Container nodes

Containers are SBGN constructions that contain one or several other SBGN constructs. Right now, *compartment* is the only container node in SBGN Activity Flow Level 1.

2.5.1 Glyph: *Compartment*

In order to describe biochemical and cellular events, it is useful to define the notion of pools. A pool is an ensemble of participants that can be considered to be identical for the events in which they are involved. A compartment is a logical or physical structure that contains pools. A pool can only belong to one compartment. Therefore, the “same” biochemical species located in two different compartments are in fact two different pools.

SBO Term:

SBO:0000289 ! functional compartment

Container:

A compartment is represented by a surface enclosed in a continuous border or located between continuous borders. These borders should be noticeably thicker than the borders of the ANs. A compartment can take **any** geometry. A compartment must always be entirely enclosed.

Label:

The identification of the compartment is carried by an unbordered box containing a string of characters. The characters can be distributed on several lines to improve readability, although this is not mandatory. The label box can be attached anywhere in the container box. Note that the label can spill-over from the container box.

Auxiliary items:

A *compartment* can carry a certain number of *units of information*, that will add information for instance about the physical environment, such as pH, temperature or voltage, see Section 2.4.1. The center of the bounding box of a *unit of information* is located on the mid-line of the border of the compartment.

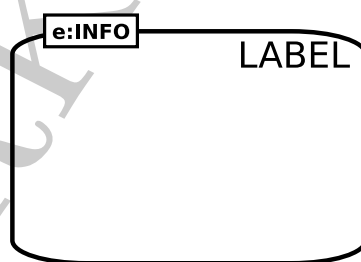


Figure 2.8: The Activity Flow glyph for compartment.

It is important to note that a compartment never contains another compartment, but may surround it. A key aspect of correctly drawing two “adjacent” compartments is that they are not separated by one line, but by **two** lines. Figure 2.9 on the following page provides an example of this in which a cell is shown made up of a nucleus surrounded by the cytoplasm.

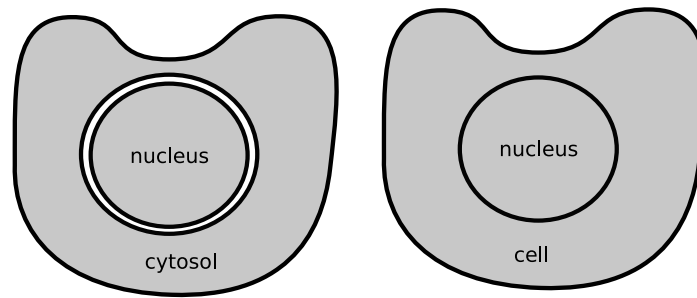


Figure 2.9: *Compartments can surround other compartments; in that case, both of the compartment's borders must still be shown, with the result that the separation is drawn as two lines. The left example is correct, with two disjoint compartments representing the “cytoplasm” and the “nucleus”. The right example is incorrect. Indeed the compartments “cell” and “nucleus” would be disjoint, the latter only overlapping the former. As a result, the volume of the nucleus is duplicated.*

The example diagram in Figure 2.10 represents three adjacent compartments. Two of the compartments carry units of information. Notice that these units of information do not overlap multiple membrane boundaries.

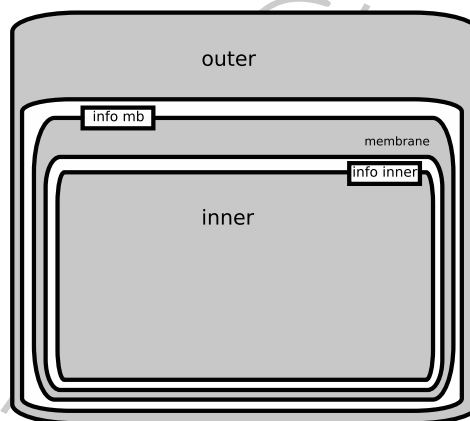


Figure 2.10: *Illustration of units of information and surrounding compartments.*

To allow more aesthetically pleasing and understandable diagrams, compartments are allowed to overlap each other visually, but it must be kept in mind that this does not mean the top compartment contains part of the bottom compartment. Figure 2.11 on the following page shows two semantically equivalent placement of compartments:

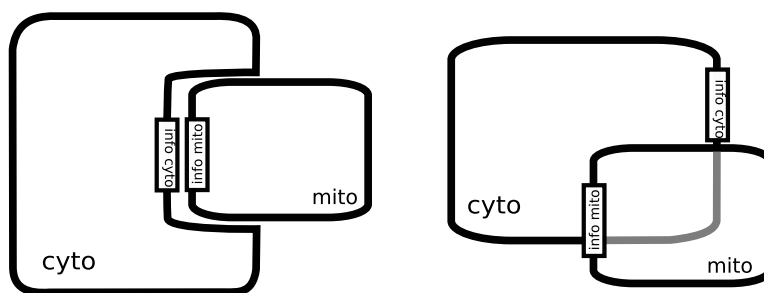


Figure 2.11: *Overlapped compartments are permitted, but the overlap does not imply containment.*

Overlapped (hidden) part of the compartment should not contain any object which could be covered by an overlapping compartment. Figure 2.12 illustrates the problem using an incorrect diagram.

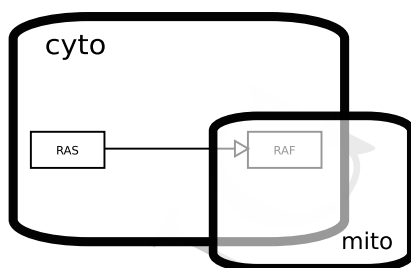


Figure 2.12: *Example of an **incorrect** diagram. Overlapped compartments must not obscure other objects.*

2.6 Glyph: Submap

A *submap* is used to encapsulate processes (including all types of nodes and edges) within one glyph. The submap hides its content to the users, and display only input terminals (or ports), linked to *ANs* (Section ??). A submap is not equivalent to an omitted process (see Section ??). In the case of an SBGN diagram that is made available through a software tool, the content of a submap may be available to the tool. A user could then ask the tool to expand the submap, for instance by clicking on the icon for the submap. The tool might then expand and show the submap within the same diagram (on the same canvas), or it might open it in a different canvas.

SBO Term:

To be determined.

Container:

The *submap* is represented as a rectangle box to remind the viewer that it is fundamentally a biological activity node.

Label:

The identification of the *submap* is carried by an unbordered box containing a string of characters. The characters may be distributed on several lines to improve readability,

although this is not mandatory. The label box has to be attached to the center of the container box.

Auxiliary items:

A *submap* carries labeled terminals. When the submap is represented folded, those terminals are linked to external ANs (Section ??). In the unfolded view, exposing the internal structure of the submap, a set of *tags* point to the corresponding internal ANs Section ??.

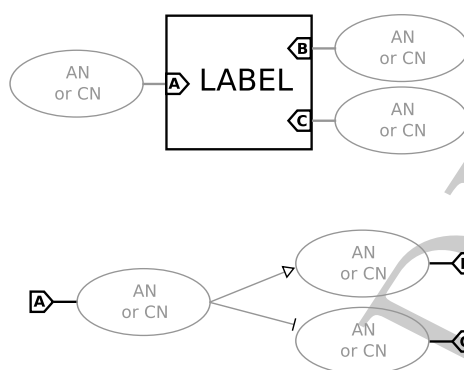


Figure 2.13: The Process Diagram glyph for submap. (Upper part) folded submap. (Lower part) content of the submap.

Figure 2.14 represents a *submap* of inhibitory G-protein coupled receptor signaling. The *submap* carries five terminals, three linked to biological ANs, and two linked to *compartments*. Note that the terminals do not define a “direction”, such as input or output. The flux of the reactions is determined by the context.

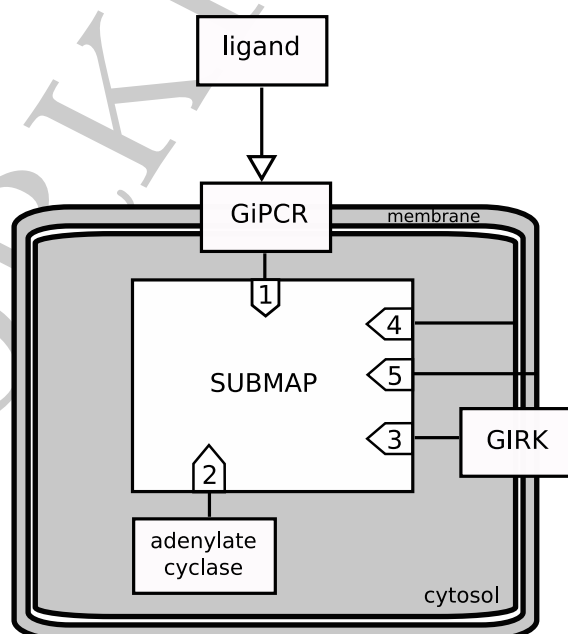


Figure 2.14: Example of a submap with contents elided.

The diagram in Figure 2.15 represents an unfolded version of a submap. Here, anything outside the submap has disappeared (e.g., ligand in Figure 2.14 on the preceding page), and the internal *tags* are not linked to the corresponding external *terminals*.

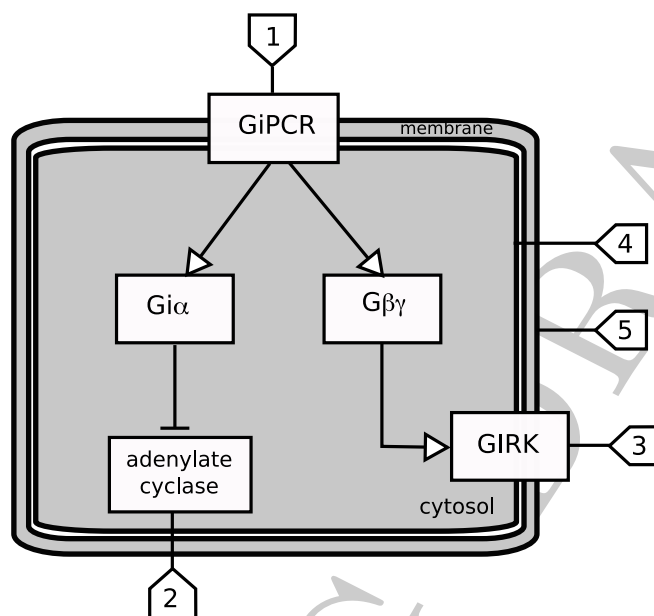


Figure 2.15: Example of an unfolded submap. The unfolded submap corresponds to the folded submap of Figure 2.14 on the preceding page.

2.7 Modulation arcs

Modulation arcs are lines that link ANs together. The symbols attached to their end extremities indicate their semantics. The modulation arcs can be used to represent direct influence from one activity to another, such as nicotine to nicotinic acetylcholine receptor activity, or indirect influence, such as

2.7.1 Glyph: *Positive influence*

In SBGN Activity Flow Level 1, a *positive influence* is defined as an action that produces positive or activating effect from one activity to another.

SBO Term:

SBO:0000170 ! stimulation

Origin:

Any *Activity node* (Section 2.3) or any *logical operator* (Section 2.8).

Target:

Any *biological activity node* or *phenotype* (Section 2.3).

End point:

The target extremity of a *positive influence* carries an open arrow pointing to the target activity node (Figure 2.16 on the next page).

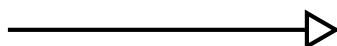


Figure 2.16: *The Activity Flow glyph for positive influence.*

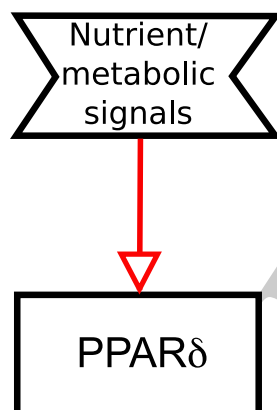


Figure 2.17: *An example of positive influence from a perturbation to the nuclear hormone receptor PPAR δ .*

2.7.2 Glyph: *Negative influence*

A *negative influence* is defined as an action that produces a negative or inhibiting effect from one activity to another.

SBO Term:

SBO:0000169 ! inhibition

Origin:

Any *Activity node* (Section 2.3) or any *logical operator* (Section 2.8).

Target:

Any *Activity node* (Section 2.3).

End point:

The target extremity of a *negative influence* carries a bar perpendicular to the arc (Figure 2.18).



Figure 2.18: *The Activity Flow glyph for negative influence.*



Figure 2.19: An example of negative influence from "Twist-1" activity to "PGC-1α" activity.

2.7.3 Glyph: *Unknown influence*

An *unknown influence* is used when the effect exerted from one activity to another is not well understood.

SBO Term:

SBO:0000168 ! control

Origin:

Any *Activity node* (Section 2.3) or any *logical operator* (Section 2.8).

Target:

Any *Activity node* (Section 2.3).

End point:

The target extremity of a *unknown influence* carries an open diamond (Figure 2.20).



Figure 2.20: The Activity Flow glyph for unknown influence.

2.7.4 Glyph: *Necessary stimulation*

A *necessary stimulation* is an influence that has to be present for the target activity to take place (to become true). An activity modulated by a necessary stimulation can only exist when this stimulation is true, whatever are the other influences this activity is subjected to.

SBO Term:

SBO:0000171 ! necessary stimulation

Origin:

Any *Activity node* (Section 2.3) or any *logical operator* (Section 2.8).

Target:

Any *Activity node* (Section 2.3).

End point:

The target extremity of a *necessary stimulation* carries a perpendicular bar followed by an open arrow pointing to the target activity node (Figure 2.21 on the following page).

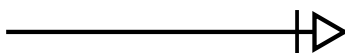


Figure 2.21: *The Activity Flow glyph for necessary stimulation.*

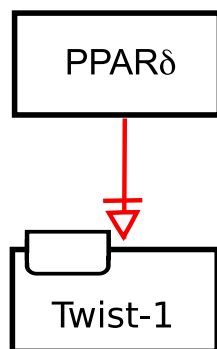


Figure 2.22: *An example of necessary stimulation where nuclear hormone receptor PPAR δ transcription factor activity stimulates the gene expression of Twist-1.*

2.8 Logical operators

2.8.1 Glyph: *And*

The glyph *and* is used to denote that all the ANs linked as input are necessary to influence the target activity.

SBO Term:

SBO:0000173 ! and.

Origin:

More than one AN (Section 2.3) or logical operator (section 2.8).

Target:

Modulation arc (Section 2.7).

Node:

And is represented by a circle carrying the word “AND”.

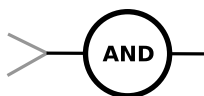


Figure 2.23: *The Activity Flow glyph for and. Only two inputs are represented, but more would be allowed.*

2.8.2 Glyph: *Or*

The glyph *or* is used to denote that any of the ANs linked as input is sufficient to influence the target activity.

SBO Term:

SBO:0000174 ! or.

Origin:

More than one AN (section 2.3) or logical operator (section 2.8).

Target:

A modulation arc (section 2.7).

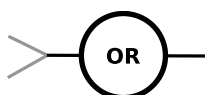
Node:*Or* is represented by a circle carrying the word “OR”.

Figure 2.24: The Activity Flow glyph for or. Only two inputs are represented, but more would be allowed.

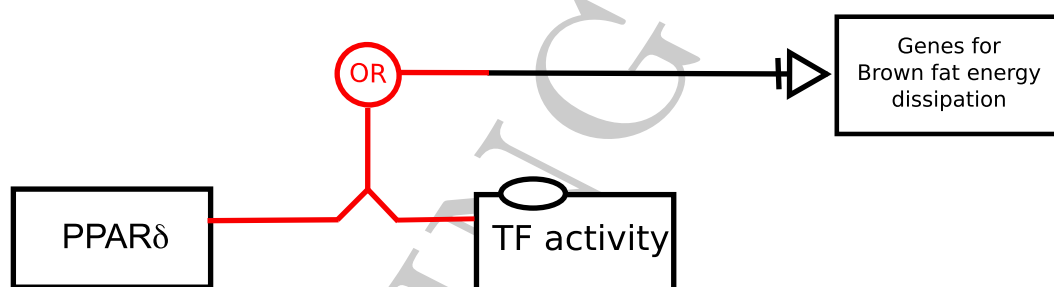


Figure 2.25: An example of the or logic operator, where the “Genes for brown fat energy dissipation” is stimulated by either the PPAR delta activity or an unspecified transcription factor activity.

2.8.3 Glyph: Not

The glyph *not* is used to denote that the AN linked as input cannot influence the target activity.

SBO Term:

SBO:0000238 ! not.

Origin:

One AN (section 2.3) or logical operator (section 2.8).

Target:

A modulation arc (section 2.3).

Node:*Not* is represented by a circle carrying the word “NOT”.

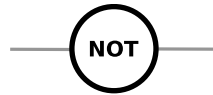


Figure 2.26: *The Activity Flow glyph for not.*

2.8.3.1 Glyph: delay

The glyph *delay* is used to denote that the *activity node* linked as input does not produce the influence immediately.

SBO Term:

SBO:NEW ! delay.

Container:

Delay is represented by a circle

Label:

Delay is identified by the greek letter “ τ ” (“TAU”) placed in an unbordered box attached to the center of the container.

Auxiliary items:

Delay does not carry any auxiliary items.

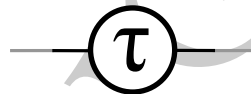


Figure 2.27: *The Entity Relationship glyph for delay.*

Chapter 3

Activity Flow Diagram grammar

3.1 Overview

3.2 Concepts

3.3 The conceptual model

3.4 Syntax

3.5 Semantic rules

3.5.1 Namespaces

The notation has a concept of a namespace within which entities with the same identifying attributes are regarded as identical. The SBGN namespaces are shown in table 3.1.

Table 3.1: *Namespace scope definitions.*

Namespace Scope	Entities affected	Notes
MapDiagram	CompartmentNode, SubMapDiagram, EquivalenceNode	
CompartmentShape	BasicEntityNode	If no <i>compartment</i> is drawn then all BasicEntityNodes are assumed to belong to an invisible “default” compartment.
EntityType	StateVariable, Annotation	
ComplexType	EntityType	

3.5.2 Compartment spanning

In all cases an AFN cannot *belong* to more than one compartment. However, an AFN can be *drawn* over more than one compartment. In such cases the decision on which is the owning compartment is deferred to the drawing tool or the author. ComplexNodes may contain AFNs which belong to different compartments and in this way a complex can be used to describe an entity that spans more than one compartment.

This restriction makes it impossible to represent in a semantically correct way a macromolecule that spans more than one compartment — for example a receptor protein. It is clearly desirable to be able to show a macromolecule in a manner that the biologist expects (i.e., spanning from the outside through the membrane to the inside). Therefore, the author is recommended to draw the macromolecule across compartment boundaries, but the underlying SBGN semantic model will assign it to only one. The assignment to a compartment may be decided by the software drawing tool or the author. Note that this has implications for auto-layout algorithms as they will only be able to treat such entity nodes as contained within a compartment and will have no way of knowing a macromolecule spans a compartment.

The current solution is consistent with other Systems Biology representations such as SBML and BioPAX. For more information about the problems representing membrane spanning proteins and the rationale behind the current solution see Section ??.

3.5.3 Compartments

The layout of compartments in an SBGN diagram does not imply anything about the topology of compartments in the cell. Compartments should be bounded and may overlap. However, adjacency and the nesting of compartments does not imply that these compartments are next to each other physically or that one compartment contains the other.

3.5.4 Modulation

It is implied, but not defined explicitly that a process has a rate at which it converts its input AFNs to its output AFNs. This concept is important in understanding how SBGN describes process modulation.

1. A process with no modulations has an underlying “basal rate” which describes the rate at which it converts inputs to outputs.
2. Modulation changes the basal rate in an unspecified fashion.
3. Stimulation is a modulation that’s effect is to increase the basal rate.
4. Inhibition is a modulation that’s effect is to decrease the basal rate.
5. The above types of modulation, when assigned to the same process are combined and have a multiplicative effect on the basal rate of the process.
6. Modulators that do not interact with each other in the above manner should be drawn as modulating different process nodes. Their effect is therefore additive.
7. At most one trigger can be assigned to a process. Two triggers would imply an implicit Boolean AND or OR operator. For clarity only one trigger can be assigned to a process and such combinations must be explicitly expressed as the Boolean operators.
8. At most one catalysis can be assigned to a process. A catalysis modulation implies that the exact biochemical mechanism underlying the process is known. In this context two catalysis reactions cannot be assigned to the same process as they are independent reactions. Other AFNs that modulate the catalysis can be assigned to the same process as modulators, stimulators, and inhibitors and will have a multiplicative modulation on the reaction rate defined by the catalysis.

3.5.5 Submaps

Submaps are a visual device that allow a map to be split into several views. They remain, however, part of the main map and share its namespace. As a test of validity it should be possible to reintroduce a submap into the main map by eliminating the SubMapNode and merging the equivalent EntityPoolNodes in both maps.

3.5.5.1 Rules for mapping to submaps

An ActivityFlowNode in the main map can be mapped to one in the submap using a TagNode in the submap and SubMapTerminals (see Section 2.6) in the main map. For a mapping between map and submap to exist the following must be true:

1. The identifiers in the TagNode and SubMapTerminals must be identical.
2. The AFNs must be identical.

3.5.5.2 Requirement to define a mapping

If a map and submap both contain the same AFN, then a mapping between them must be defined as above.

Chapter 4

Layout Guidelines for a Activity Flow Diagram

4.1 Introduction

The previous chapters describe the appearance and meaning of SBGN Activity Flow Level 1 components. Objects are *AFNs*, *container nodes*, *logical operators* as well as *connecting arcs*. The components of an Activity Flow diagram have to be placed in a meaningful way – a random distribution with spaghetti-like connections will most likely hide the information encoded in the underlying model, whereas an elegant placement of the objects, giving a congenial appearance of the diagrams, may reveal new insights. The arrangement of components in a diagram is called a *layout*.

SBGN Activity Flow diagrams should be easily recognisable not only by the glyphs used, but also by the general style of the layout. However, the arrangement of the components is a complex art in itself, and there is no simple rule which can be applied to all cases. Therefore this section provides guidelines for the layout of process diagrams, divided into two categories:

1. requirements, i.e., rules which **must** be fulfilled by a layout, and
2. recommendations, i.e., rules which **should** be followed if possible.

In addition, we provide a list of additional suggestions which may help in producing aesthetically more pleasant layouts, possibly easier to understand.

Those layout guidelines are independent of the method used to produce the diagram, and apply to both manually drawn diagrams as well as diagrams produced by an automatic layout algorithm. The guidelines do not deal with interactive aspects (e.g., the effect of zooming). Further information about automatic network layout (graph drawing) can be found, for example, in the books of Di Battista and co-authors [14] and Kaufmann and Wagner [15].

Please note that the color of objects do not carry any meaning in SBGN. Although one can use colors to emphasize part of a diagram or encode additional information, the meaning of the diagram should not depend on the colors. Furthermore, objects can have different sizes and size is also meaningless in SBGN. For example, a transition node may be larger than a protein node. Also the meaning of a graph should be conserved upon scaling as far as possible.

4.2 Layout guidelines

4.2.1 Requirements

Requirements are rules which **must** be fulfilled by a layout to produce a valid SBGN Activity Flow Level 1 diagram.

4.2.1.1 Node-node overlaps

Nodes are only allowed to overlap in two cases:

1. the overlapping nodes define a glyph (e.g., a stacking of *biological activity* and *unit of information*).
2. nodes overlapping compartments (e.g., a complex placed on the compartment border).

Otherwise, nodes are not allowed to overlap (Figure 4.1). This includes the touching of nodes, which is also not allowed. Submaps are not allowed to overlap.

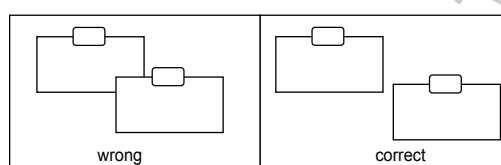


Figure 4.1: Nodes must not overlap.

4.2.1.2 Node-edge crossing

In case of node-edge crossing the edge must be drawn on the top of the node (Figure 4.2). See also recommendation 4.2.2.1 (crossing between edges and nodes should be avoided).

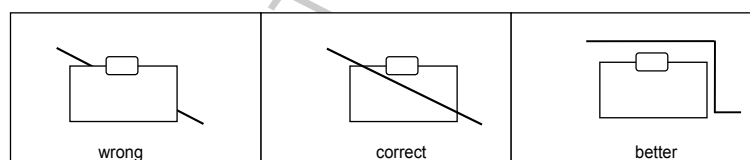


Figure 4.2: If an edge crosses a node, the edge must be drawn on top of the node.

4.2.1.3 Node border-edge overlaps

Edges are not allowed to overlap the border lines of nodes (Figure 4.3 on the next page).

4.2.1.4 Edge-edge overlaps

Edges are not allowed to overlap (Figure 4.4 on the following page). This includes touching of edges. Furthermore, an edge is neither allowed to cross itself nor to cross a boundary of node more than twice or other edges more than once.

4.2.1.5 Node orientation

Nodes have to be drawn horizontally or vertically, any other rotation of elements is not allowed (Figure 4.5 on the next page).

4.2.1.6 Node labels

At least a part of the label (unbordered box containing a string of characters) has to be placed inside the node it belongs to. Node labels are not allowed to overlap nodes or other labels (this includes touching of other nodes or labels).

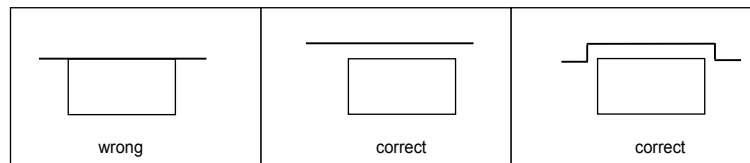


Figure 4.3: *Edges must not overlap node borders.*

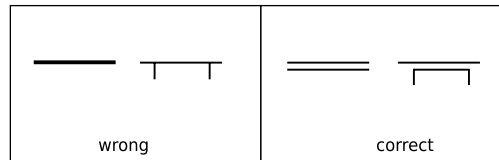


Figure 4.4: *Edges must not overlap.*

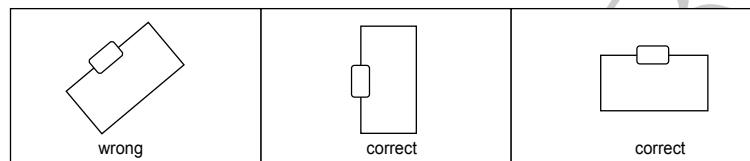


Figure 4.5: *The node orientation must be horizontally or vertically.*

4.2.1.7 Edge labels

Edge labels are not allowed to overlap nodes. This includes touching of nodes.

4.2.1.8 Compartments

If an activity has all participants in the same compartment the activity node and all edges/arcs have to be in this compartment. If an activity has participants in at least two different compartments, the activity node has to be either in a compartment where the activity has at least one participant or in the empty space.

4.2.2 Recommendations

Recommendations are rules which should be followed if possible to produce layouts may be easier to understand.

4.2.2.1 Node-edge crossing

Crossings between edges and nodes should be avoided. Some crossings may be unavoidable, e.g., the crossing between an edge and a compartment border. See also requirement 4.2.1.2 (in case of node-edge crossings the edge must be drawn on the top of the node).

4.2.2.2 Labels

Labels should be horizontal. Node labels should be placed completely inside the node if possible. Edge labels should be placed close to the edge and avoid overlapping the edge as well as other edge labels.

4.2.2.3 Avoid edge crossings

The amount of crossings between edges should be minimized.

4.2.3 Additional suggestions

Here is a list of additional layout suggestions which may help in producing aesthetically more pleasing layouts which may be easier to understand.

- Angle of edge crossings: If edge crossings are not avoidable edges should cross with an angle close to 90 degrees.
- Drawing area and width/height ratio: The drawing should be compact and the ratio between the width and the height of the drawing should be close to 1.
- Edge length: Long edges should be avoided.
- Number of edge bends: Edges should be drawn with as few bends as possible.
- Similar and symmetric parts: Similar parts of a diagram should be drawn in a similar way, and symmetric parts should be drawn symmetrically.
- Proximity information: Related elements (e.g., nodes connected by edges within a compartment) should be drawn close together.
- Directional information: Subsequent processes (e.g., a sequence of activities) should be drawn in one direction (e.g., from top to bottom or from left to right).
- Compartments: Different compartments should have different background shade or color.

Appendix A

Examples

This section is still an on going effort. More examples will be added.

The following diagrams present examples of SBGN Activity Flow diagrams representing biological activities and their influences among each other in pathway networks. They by no mean exhaust the possibilities of SBGN Activity Flow Level 1.

Figure A.1 presents an example of a signaling pathway involving the regulation of TGF β -induced metastasis. The pathway was described in a report titled "A Mutant-p53/Smad Complex Opposes p63 to Empower TGF β -Induced Metastasis" in the April issue of Cell [16]. The figure shows the usage of *biological activity nodes*, *phenotype*, *positive influence arc*, *negative influence arc*, *necessary stimulation arc*, and *logic operator*.

Figure A.2 on the next page presents a more complicated example of signaling pathway involving the intracellular signaling through the epidermal growth factor receptor (EGFR).

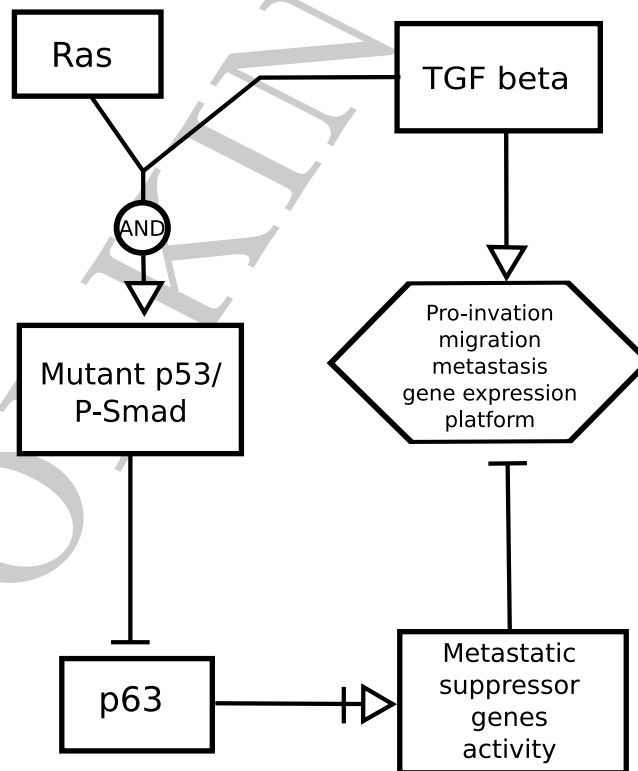


Figure A.1: Regulation of TGF β -induced metastasis.

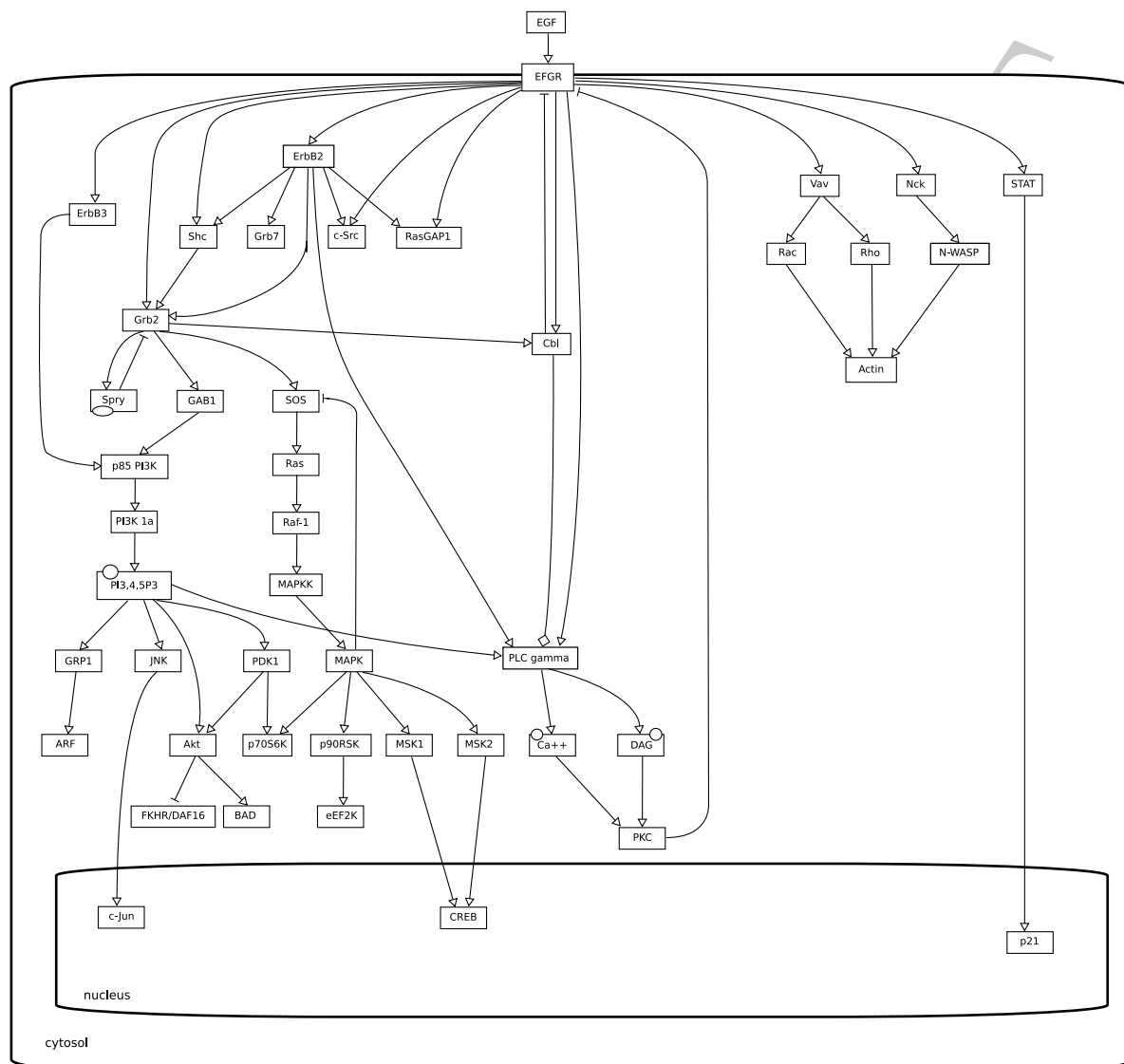


Figure A.2: Epidermal Growth Factor Receptor Pathway.

This example is a redraw of the Epidermal Growth Factor Receptor Pathway described in the Signal Transduction Knowledge Environment (http://stke.sciencemag.org/cgi/cm/stkecm;CMP_14554).

Figure A.3 on the following page shows the transforming growth factor beta ($TGF\beta$) signaling pathway. The map is a redraw of the TRG-beta Signaling Pathway described in the PANTHER Pathway System (<http://www.pantherdb.org/pathway/pathwayDiagram.jsp?catAccession=P00052>) and is based on reviews by Massague [17] and Derynck [18].

Figure A.4 on page 32 presents the simplest view of action potential propagation mediated by the voltage-gated sodium channels. There are two views how voltage-gated sodium channels are involved. The diagram on the left side shows that the *increase in membrane potential* activates *voltage-gated sodium channel activity*, which in turn triggers *membrane depolarization*. The diagram on the right side provides more detail in the mechanism. It shows that the *increase in membrane potential* first activates the *gating activity* of the channel, which in turn activates the *conductance activity* leading to the *membrane depolarization*. In this case, both *gating activity* and *conductance activity* come from the sodium channel gene, which is indicated as

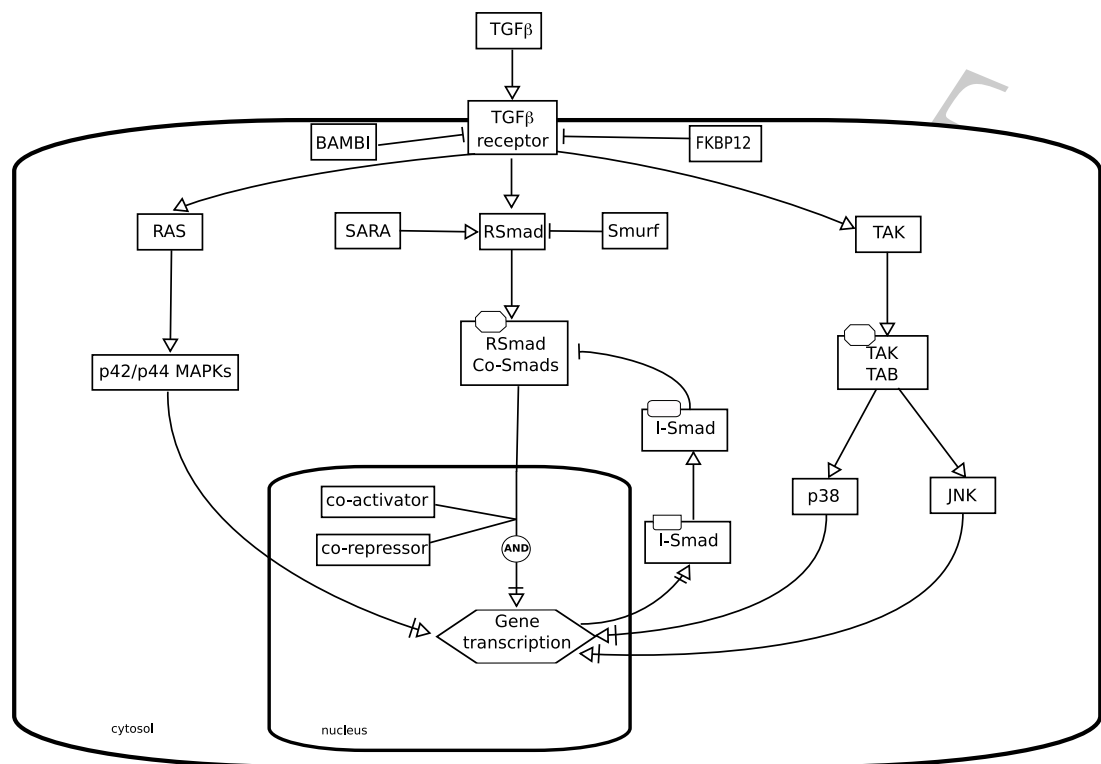


Figure A.3: Transforming Growth Factor beta signaling pathway.

the *unit of information*. In addition, this example also shows the advantage of using Activity Flow Diagram, because certain activities, such as gating and conductance, come from a number of amino acids in particular three dimensional structure that are not able to be illustrated in either Process Diagrams or Entity Relationship Diagrams

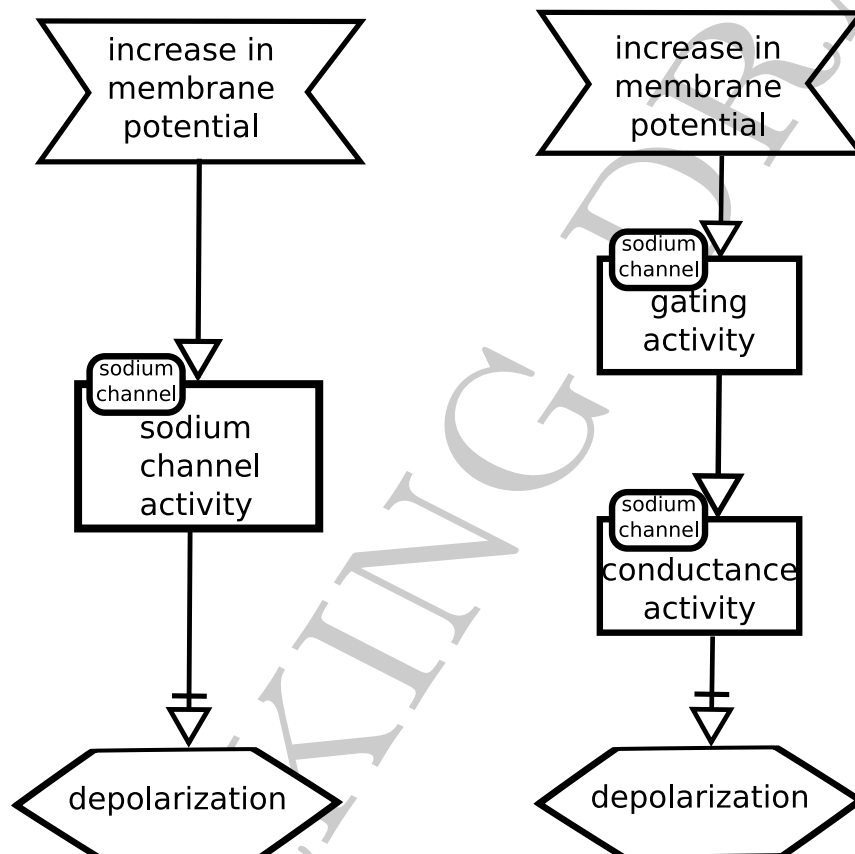


Figure A.4: Two views of the role voltage-gated sodium channel plays in action potential generation illustrated by SBGN Activity Flow Level 1.

Appendix B

Reference card

Print this summary of SBGN Activity Flow symbols for a quick reference.

WORKING DRAFT

negative influenza

Target PN

unknown influenza



Bibliography

- [1] G. Michal. On representation of metabolic pathways. *BioSystems*, 47:1–7, 1998.
- [2] K. W. Kohn. Molecular interaction map of the mammalian cell cycle control and DNA repair systems. *Molecular Biology of the Cell*, 10(8):2703–2734, 1999.
- [3] P. P.-S. Chen. The entity-relationship model – toward a unified view of data. *ACM Transactions on Database Systems*, 1(1):9–36, 1976.
- [4] I. Pirson, N. Fortemaïson, C. Jacobs, S. Dremier, J. E. Dumont, and C. Maenhaut. The visual display of regulatory information and networks. *Trends in Cell Biology*, 10(10):404–408, 2000.
- [5] D. L. Cook, J. F. Farley, and S. J. Tapscott. A basis for a visual language for describing, archiving and analyzing functional models of complex biological systems. *Genome Biology*, 2(4):R0012.1–R0012.10, 2001.
- [6] E. Demir, O. Babur, U. Dogrusoz., A. Gursoy, G. Nisanci, R. Cetin-Atalay, and M. Ozturk. Patika: an integrated visual environment for collaborative construction and analysis of cellular pathways. *Bioinformatics*, 18(7):996–1003, 2002.
- [7] E. Demir, O. Babur, U. Dogrusoz, A. Gursoy, A. Ayaz, G. Gulesir, G. Nisanci, and R. Cetin-Atalay. An ontology for collaborative construction and analysis of cellular pathways. *Bioinformatics*, 20(3):349–356, 2004.
- [8] R. Maimon and S. Browning. Diagrammatic notation and computational structure of gene networks. In H. Kitano, editor, *Proceedings of the 2nd International Conference on Systems Biology*, pages 311–317, Madison, WI, 2001. Omnipress.
- [9] H. Kitano. A graphical notation for biochemical networks. *BioSilico*, 1:169–176, 2003.
- [10] H. Kitano, A. Funahashi, Y. Matsuoka, and K. Oda. Using process diagrams for the graphical representation of biological networks. *Nature Biotechnology*, 23(8):961–966, 2005.
- [11] S. L. Moodie, A. A. Sorokin, I. Goryanin, and P. Ghazal. A graphical notation to describe the logical interactions of biological pathways. *Journal of Integrative Bioinformatics*, 3:36.1–36.11, 2006.
- [12] M. Hucka, A. Finney, H. M. Sauro, H. Bolouri, J. C. Doyle, H. Kitano, A. P. Arkin, B. J. Bornstein, D. Bray, A. Cornish-Bowden, A. A. Cuellar, S. Dronov, E. D. Gilles, M. Ginkel, V. Gor, I. I. Goryanin, W. J. Hedley, T. C. Hodgman, J.-H. Hofmeyr, P. J. Hunter, N. S. Juty, J. L. Kasberger, A. Kremling, U. Kummer, N. Le Novère, L. M. Loew, D. Lucio, P. Mendes, E. Minch, E. D. Mjolsness, Y. Nakayama, M. R. Nelson, P. F. Nielsen, T. Sakurada, J. C. Schaff, B. E. Shapiro, T. S. Shimizu, H. D. Spence, J. Stelling, K. Takahashi, M. Tomita, J. Wagner, and J. Wang. The Systems Biology Markup Language (SBML): A medium for representation and exchange of biochemical network models. *Bioinformatics*, 19(4):524–531, 2003.

- [13] D. Pan, M. Fujimoto, A. Lopes, and Y. Wang. Twist-1 is a ppar δ -inducible, negative-feedback regulator of pgc-1 α in brown fat metabolism. *Cell*, 137(1):73–86, 2009.
- [14] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall, New Jersey, 1998.
- [15] M. Kaufmann and D. Wagner. *Drawing Graphs: Methods and Models*, volume 2025 of *Lecture Notes in Computer Science Tutorial*. Springer, 2001.
- [16] M. Adorno, M. Cordenonsi, M. Montagner, S. Dupont, C. Wong, B. Hann, A. Solari, S. Bobisse, M. B. Rondina, V. Guzzardo, A. R. Parenti, A. Rosato, S. Bicciato, A. Balmain, and S. Piccolo. A mutant-p53/smad complex opposes p63 to empower tgfbeta-induced metastasis. *Cell*, 137(1):87–98, 2009.
- [17] J. Massague. Tgf-beta signal transduction. *Annu Rev Biochem*, 67:753–791, 1998.
- [18] R. Derynck, R. J. Akhurst, and A. Balmain. Tgf-beta signaling in tumor suppression and cancer progression. *Nature Genetics*, 29(2):117–129, 2001.