BioTapestry User's Guide

Revision A

August 16, 2005

William Longabaugh

Bolouri Group

Institute for Systems Biology

Table of Contents

Frequently Asked Questions

Here are some pointers into this User's Guide for some common questions.

What's a fast way to learn how to use BioTapestry?

You can download the separate BioTapestry tutorial, which takes you step-by-step through building a small network, including a dynamic model that is based upon experimental data.

Is there an easy way to build a set of submodels without having to build tables of experimental data?

Instead of creating dynamic hourly submodels, you can build static submodels. While they are less interactive (no hourly time slider), they are easier to build; you just click on elements to include in the model. The BioTapestry tutorial has a section on creating static submodels; they are introduced here in the section Subset Models: Static Subset Models. Instructions on how to build them are included in the section Populating Subsets: Static Subsets.

What's the relationship between all these different views of the network presented by BioTapestry?

The organization of the models in BioTapestry is discussed in the section on The Model Hierarchy

Can I use BioTapestry without having to build a model hierarchy?

Yes, you can just draw a root model; the section entitled Drawing the Top Level Root Model describes how to do that. But if you want to introduce regions and/or have pieces of the network duplicated, you will need to build a top-level instance model from the root model. And if you want to show how the network evolves over time, you will need to build static or dynamic submodels. These topics are covered in Creating Submodel Instances and Populating Submodels.

Do I have to enter all the experimental data by hand?

No, data can be imported via data files. For importing QPCR perturbation data, go to the section on Importing Experimental Data. For importing time course and temporal input data, take a look at the sample XML data files that accompany the BioTapestry tutorial.

I don't like the colors assigned by BioTapestry to the regions, genes, and links. How can I change them?

You can change the colors of network elements by right-clicking on them choosing the **Properties...** item in the menu, which brings up the Properties dialog. Colors are assigned on the **Presentation Properties** tab. BioTapestry comes with a set of pre-defined colors that you can assign to network elements, and you can edit the definitions of (most of) those colors, as well as define new colors. Editing properties is covered in the section Editing Properties.

Why won't BioTapestry allow me to draw a link a certain way?

There are some rules in BioTapestry about how links are created and modified. These rules are covered in the sections on Drawing Links and Editing Links.

How do I get model diagrams to publish?

While viewing any model in the network hierarchy, you can select **File->Export->Export Image...** from the main menu and output PNG or JPEG images of any desired resolution.

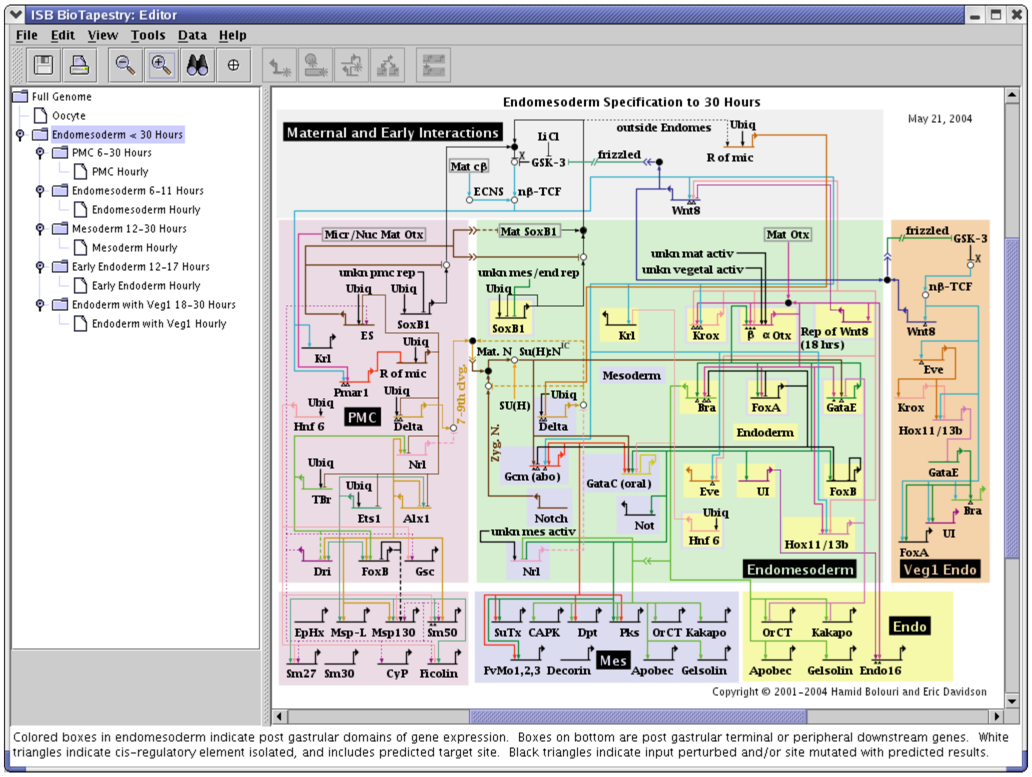
How do I create embedded regions like the Mesoderm and Endoderm in the sea urchin endomesoderm network?

The section on Setting Up Subregions describes this feature.

If you have any other questions that you feel belong in this FAQ, let us know at:

`biotapestry at systemsbiology dot org`

Introduction



Colored boxes in endomesoderm indicate post gastrular domains of gene expression. Boxes on bottom are post gastrular terminal or peripheral downstream genes. White triangles indicate cis-regulatory element isolated, and includes predicted target site. Black triangles indicate input perturbed and/or site mutated with predicted results.

BioTapestry is an interactive tool for building, visualizing, and simulating genetic regulatory networks. It is designed around the concept of a developmental network model, and is intended to deal with large scale models with consistency and clarity. It is capable of representing systems that exhibit increasing complexity over time, such as the genetic regulatory network controlling endomesoderm development in sea urchin embryos. Significant features include:

Network models can either be created by explicitly drawing them, or they can be generated and automatically laid out from lists of interactions. These lists of interactions can be provided either through interactive dialogs or comma-separated value (CSV) files exported by spreadsheet programs.

A set of tools is available to help the user draw and reorganize network layouts, including automatic link layout support, network expansion and compression, layout synchronization, and so forth.

Supporting data resulting from the perturbation of expression of specific genes, measured in any way (QPCR, genetics, etc.), can be easily accessed for each gene. Temporal and spatial expression results are also accessible.

Regulatory state models can be automatically driven by the temporal and spatial expression and input data tables for each network element. This feature allows BioTapestry to present hourly views of the evolving network.

BioTapestry is being integrated with the other tools in the ISB Bolouri Lab MAGNET suite.

Additional features are also under development. The next BioTapestry version (anticipated in Fall 2005) will analyze experimental spatial/temporal expression data and raw perturbation data, and then help the user decide what network connections should be included in the network diagram.

This user's guide will first describe the organization of the network model. Then there will be a discussion of how to build models using drawing techniques, followed by a discussion of how to build models using interaction tables. These topics will be covered:

1) Drawing genes, nodes, and links in the root model.

2) Creating and populating submodels.

3) What experimental data is supported.

4) Building up the experimental datasets.

5) Details about setting up subregions.

6) Editing properties of model elements.

7) How to build networks using interaction tables.

8) Features of the automatic layout engine.

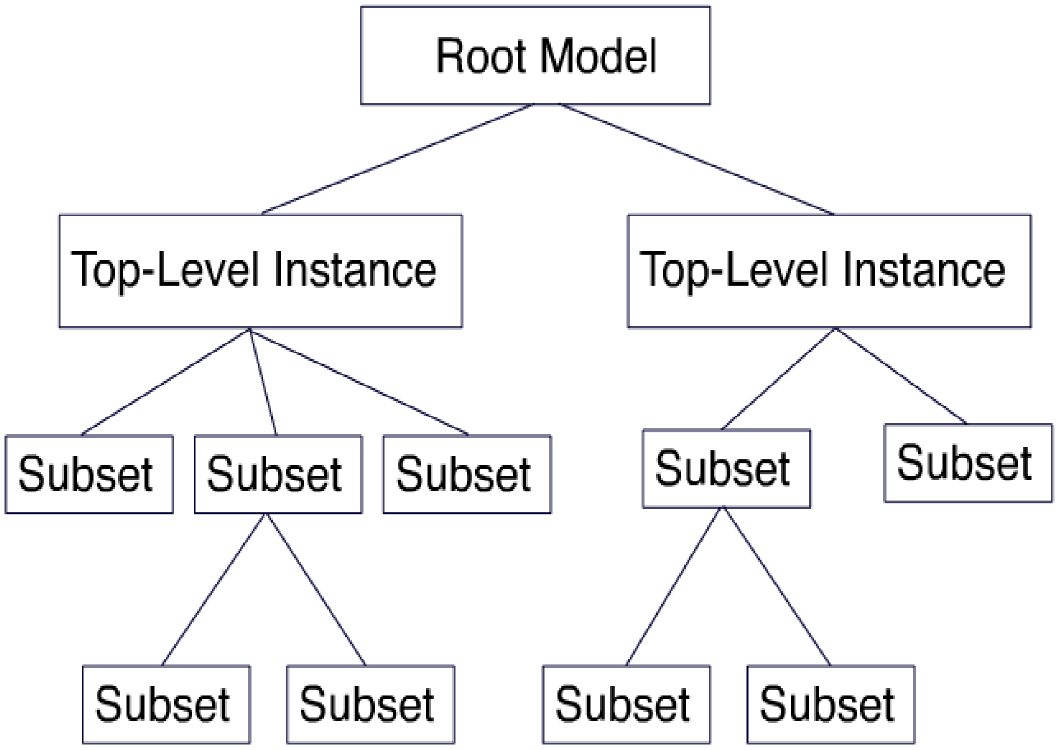
9) Using comma-separated value files to create full model hierarchies.

10) An overview of command-line arguments.

This guide will then conclude with an overview of some of BioTapestry's import and export capabilities.

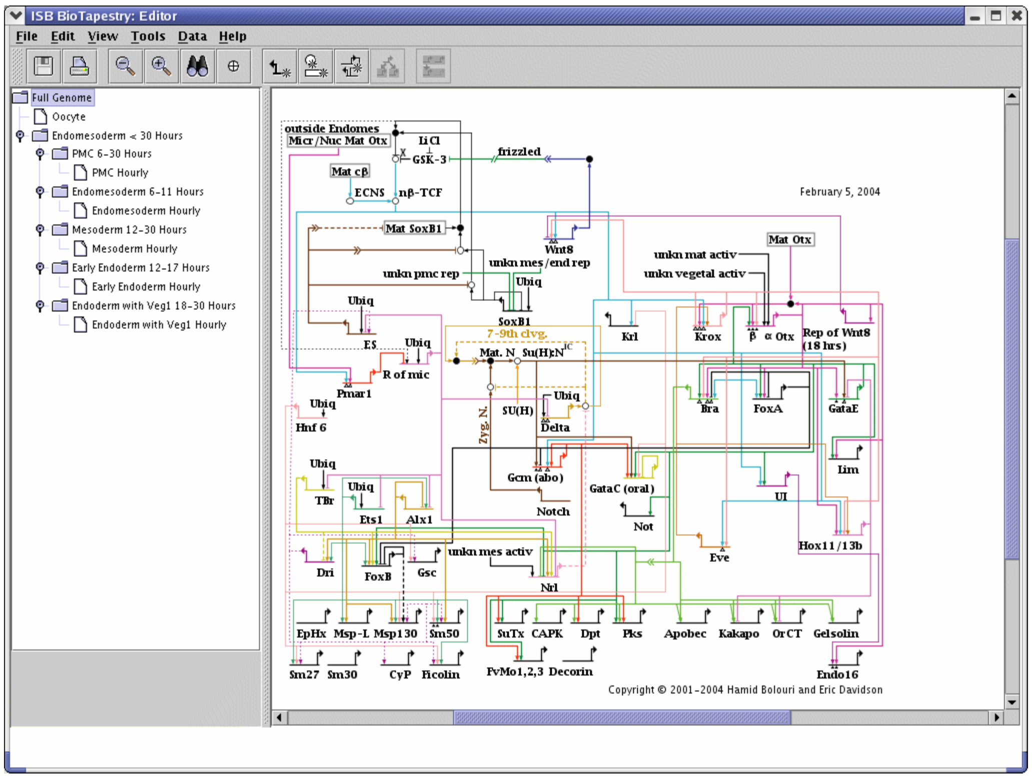
The Model Hierarchy

BioTapestry organizes a network model into a layered, multi-level hierarchy consisting of a single **root model** at the top, zero or more **top-level instance models** in the next layer, and then zero or more additional layers consisting of any number of **subset models**:



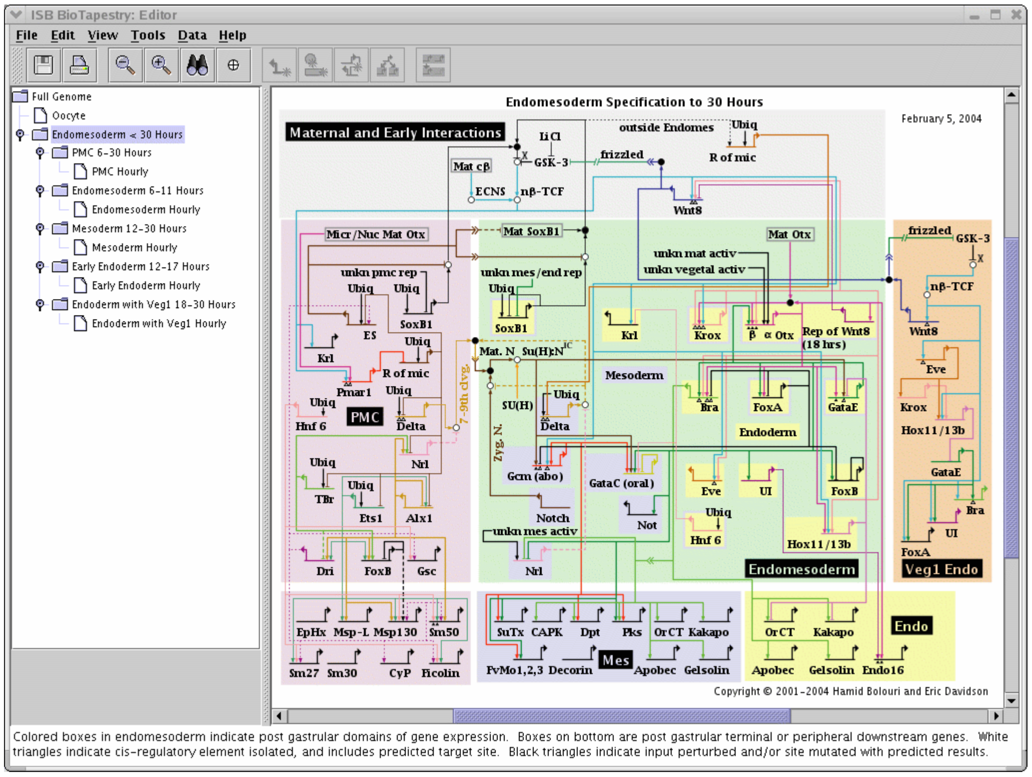
The Top-Level Root Model

At the top of the hierarchy sits the root model; all submodels are derived from this model. The root model contains a single copy of each gene, node or link that appears in any of the submodels. Thus, this level depicts all of the interactions between the network elements regardless of when and where those interactions occur:



Top-Level Instance Models

The next level in the hierarchy is where different regions are introduced. Each region can have a separate copy of any gene or node present in the root model, and there can be a unique instance of each link for each (source, destination) tuple of regions. Since this level is derived from the top-level model, consistency is assured. In the endomesoderm model example, this level is showing the interactions present in all the different regions from 6 to 30 hours. Focusing on individual regions and shorter time periods is the role of submodels further down in the hierarchy:

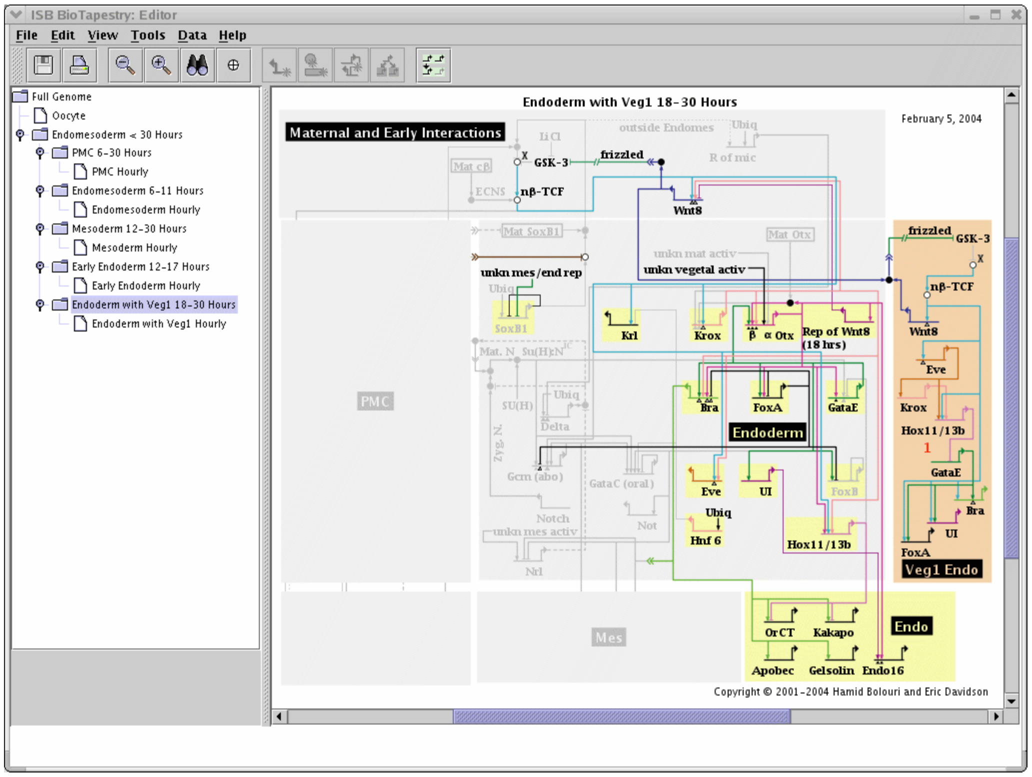

ISB BioTapestry: Editor
File Edit View Tools Data Help
Endomesoderm Specification to 30 Hours
February 5, 2004
Maternal and Early Interactions
PMC
Mesoderm
Endoderm
Endomesoderm
Veg1 Endo
Endo
Mes
Copyright © 2001-2004 Hamid Bolouri and Eric Davidson

Full Genome
- Oocyte
- Endomesoderm < 30 Hours
  - PMC 6-30 Hours
    - PMC Hourly
  - Endomesoderm 6-11 Hours
    - Endomesoderm Hourly
  - Mesoderm 12-30 Hours
    - Mesoderm Hourly
  - Early Endoderm 12-17 Hours
    - Early Endoderm Hourly
  - Endoderm with Veg1 18-30 Hours
    - Endoderm with Veg1 Hourly

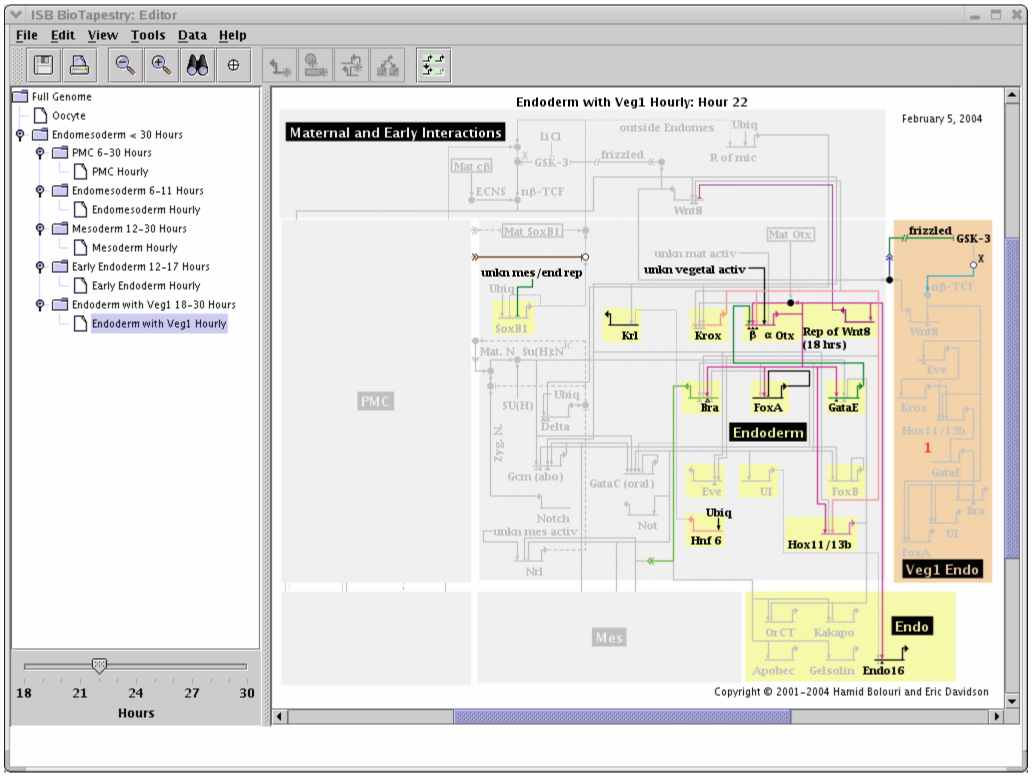
Colored boxes in endomesoderm indicate post gastrular domains of gene expression. Boxes on bottom are post gastrular terminal or peripheral downstream genes. White triangles indicate cis-regulatory element isolated, and includes predicted target site. Black triangles indicate input perturbed and/or site mutated with predicted results.

Subset Models

This level of the hierarchy shows a subset of the regions and interactions present in the parent model. In this example, this submodel includes two of the regions (the Endoderm and Veg1 Endoderm) over a reduced time span from 18 to 30 hours. For this model, BioTapestry is automatically using the underlying temporal/spatial expression data and temporal input data tables to display only those elements in these regions that are active sometime during the 12 hour period:



The lowest level of the hierarchy is again a subset of the model above it. This example shows hourly views of the Endoderm and Veg1 Endoderm. As with the parent model, these hourly views are automatically generated from underlying expression and input data tables. Each hourly view is selected using the time slider in the lower left of the BioTapestry window. This feature allows the user to quickly visualize how the network interactions evolve over time:



Static Subset Models

There are two types of subset models: static and dynamic. Static models are constructed by:

1) Choosing the regions in the parent model that you wish to include in the subset.

2) Clicking on the nodes and links that you want to include in the subset.

Dynamic Subset Models

Dynamic models use the time expression data tables and temporal input data tables to automatically populate the nodes and links in the subset model. The only action the user needs to take (other than populating the underlying data tables, and possibly specifying how network elements and regions correspond to those table entries) is to choose the regions from the parent model to include in the subset.

There are two types of dynamic subset models: summation models and hourly.

Summations

With a summation subset, you specify the hour range that the model spans, and then BioTapestry produces a single view that shows **all** of the active nodes and links present in the subset within the time period. A summation subset can still serve as a parent model to further subset models that are either summations or hourly.

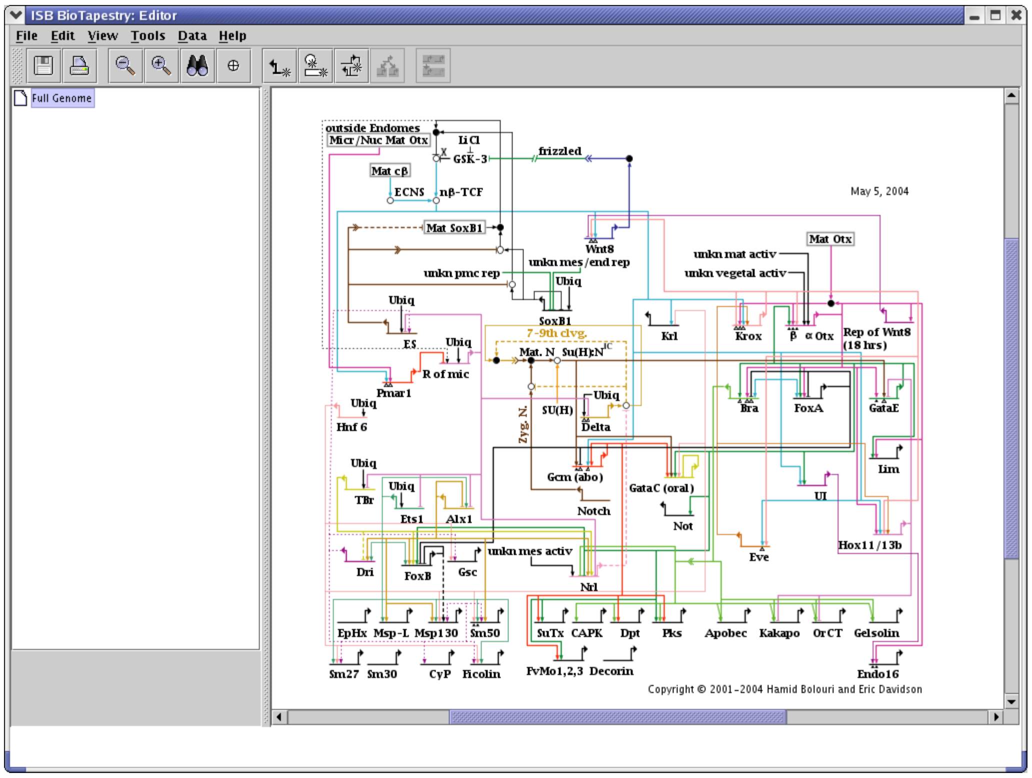
Hourly

The other type of dynamic subset is **hourly**. As with the summation subset, you specify the hours you want to display (where that range of hours is a subset of the hours in the parent model). BioTapestry then makes a set of views, one per hour, that cover this time range. Hourly subset models cannot themselves be parent models; they are required to be at the bottom of the model hierarchy.

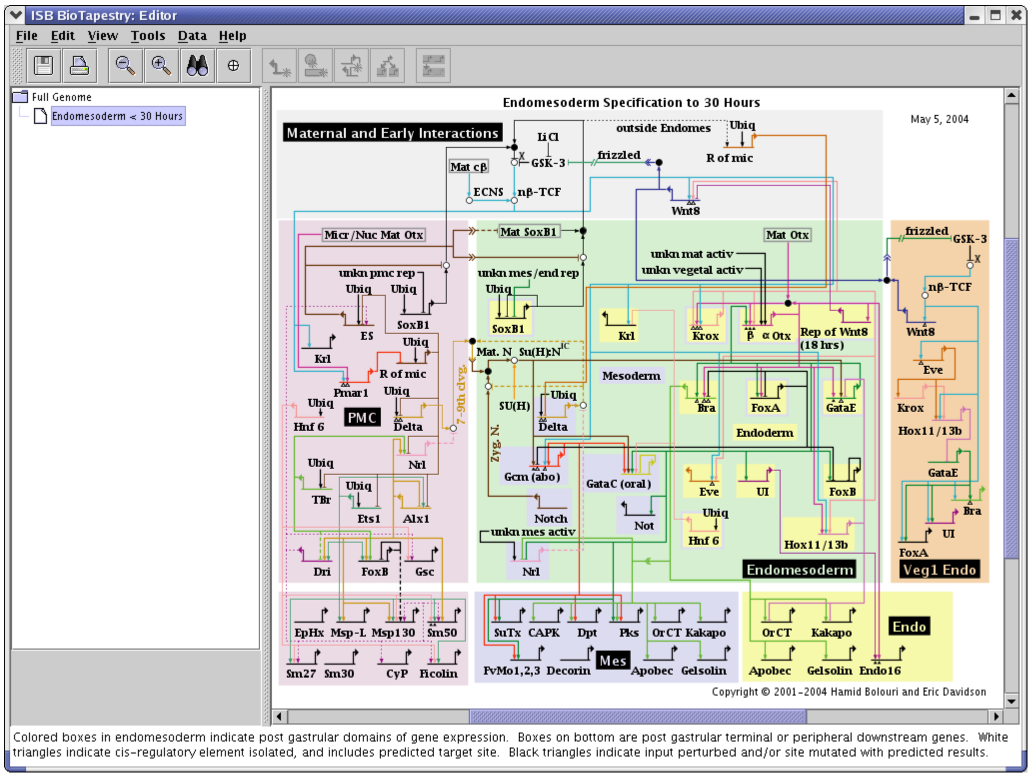
Building a Model Hierarchy

The first part of this guide deals with building a developmental model hierarchy by drawing it and propagating elements to submodels; the second part will deal with building the hierarchy with interaction tables. The drawing/propagation approach that will now be discussed involves the following steps:

1) Create the root network by drawing genes, other nodes, and links that represent interactions between these objects:



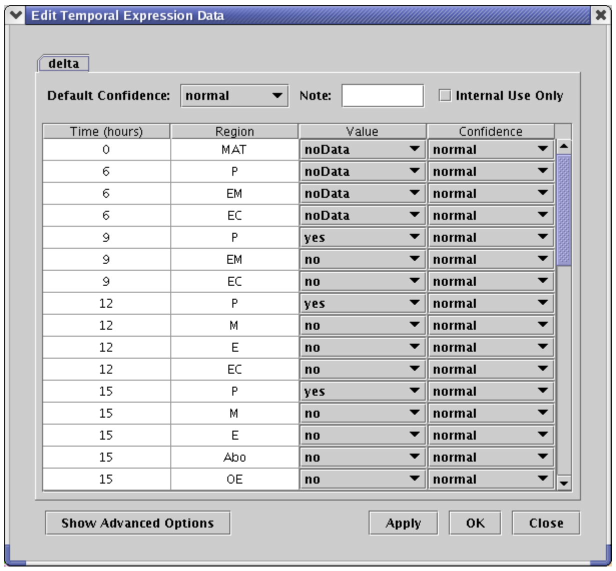
2) Create a top-level instance model as a submodel of the root, and then push down the root level elements into different regions of the instance model:



Copyright © 2001-2004 Hamid Bolouri and Eric Davidson

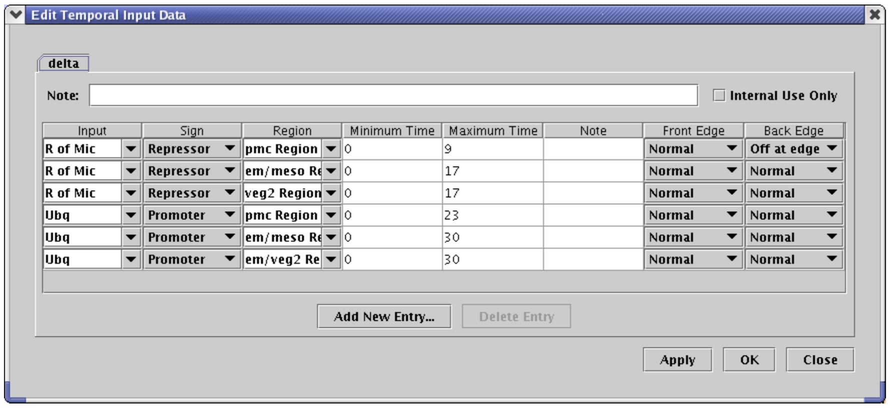
Colored boxes in endomesoderm indicate post gastrular domains of gene expression. Boxes on bottom are post gastrular terminal or peripheral downstream genes. White triangles indicate cis-regulatory element isolated, and includes predicted target site. Black triangles indicate input perturbed and/or site mutated with predicted results.

3) Enter temporal expression and temporal input data for each model element. This can be done by importing XML files, or using interactive editing dialogs:



Default Confidence: normal Note: ☐ Internal Use Only

Time (hours)	Region	Value	Confidence
0	MAT	noData	normal
6	P	noData	normal
6	EM	noData	normal
6	EC	noData	normal
9	P	yes	normal
9	EM	no	normal
9	EC	no	normal
12	P	yes	normal
12	M	no	normal
12	E	no	normal
12	EC	no	normal
15	P	yes	normal
15	M	no	normal
15	E	no	normal
15	Abo	no	normal
15	OE	no	normal


Edit Temporal Input Data

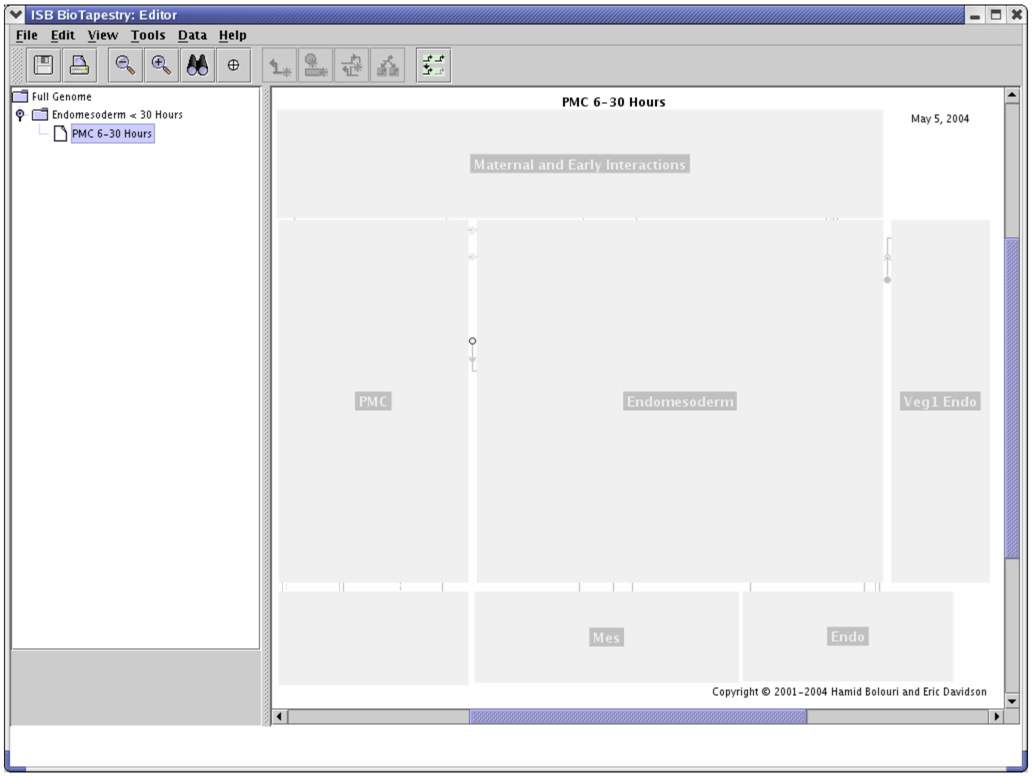
delta

Note:

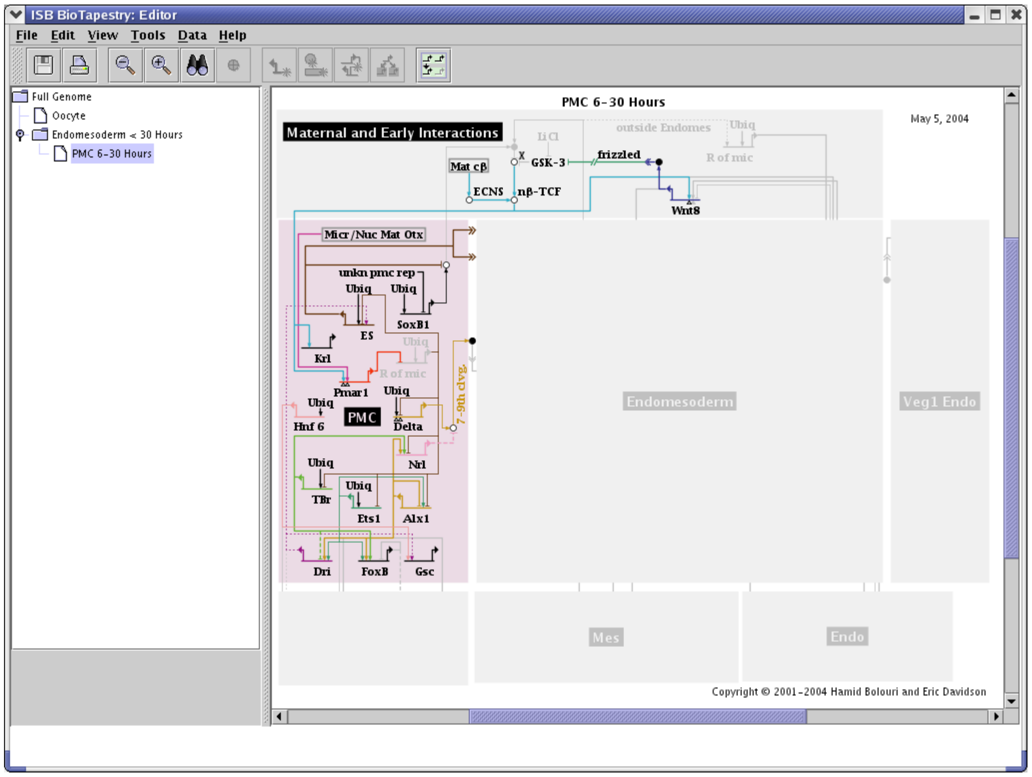
☐ Internal Use Only

Input	Sign	Region	Minimum Time	Maximum Time	Note	Front Edge	Back Edge
R of Mic	Repressor	pmc Region	0	9		Normal	Off at edge
R of Mic	Repressor	em/meso Re	0	17		Normal	Normal
R of Mic	Repressor	veg2 Region	0	17		Normal	Normal
Ubq	Promoter	pmc Region	0	23		Normal	Normal
Ubq	Promoter	em/meso Re	0	30		Normal	Normal
Ubq	Promoter	em/veg2 Re	0	30		Normal	Normal

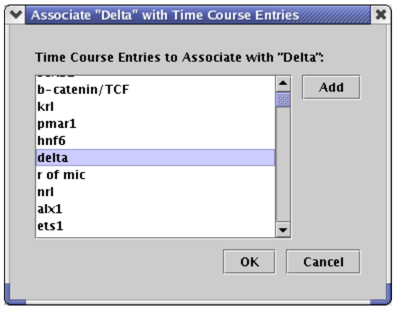
4) Create a dynamic submodel of the top-level instance model. At this point, since no regions have been pulled down from the parent model, the model is empty:



5) Finally, select the regions in the parent model to include, which produces:

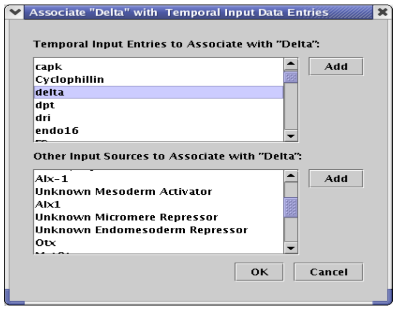


6) If the names used for data entries do not match those used in the network diagram, the user will need to set up custom data associations for nodes and regions:


Associate "Delta" with Time Course Entries

Time Course Entries to Associate with "Delta":

b-catenin/TCF
krl
pmar1
hnf6
delta
r of mic
nrl
alx1
ets1

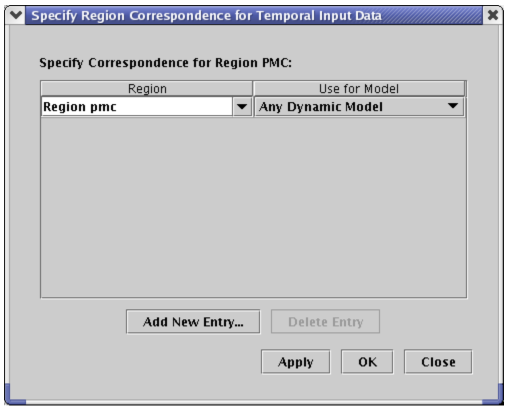

Associate "Delta" with Temporal Input Data Entries

Temporal Input Entries to Associate with "Delta":

capk
Cyclophillin
delta
dpt
dri
endo16

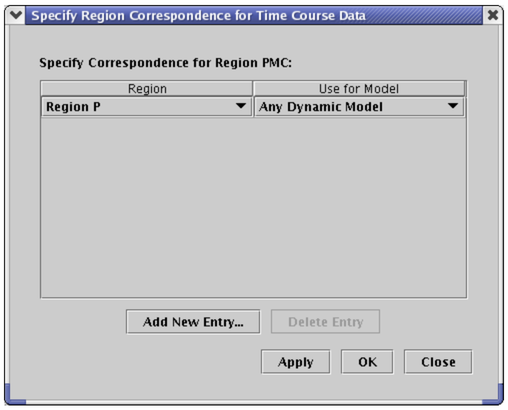
Other Input Sources to Associate with "Delta":

Alx-1
Unknown Mesoderm Activator
Alx1
Unknown Micromere Repressor
Unknown Endomesoderm Repressor
Otx


Specify Region Correspondence for Temporal Input Data

Specify Correspondence for Region PMC:

Region	Use for Model
Region pmc	Any Dynamic Model


Specify Region Correspondence for Time Course Data

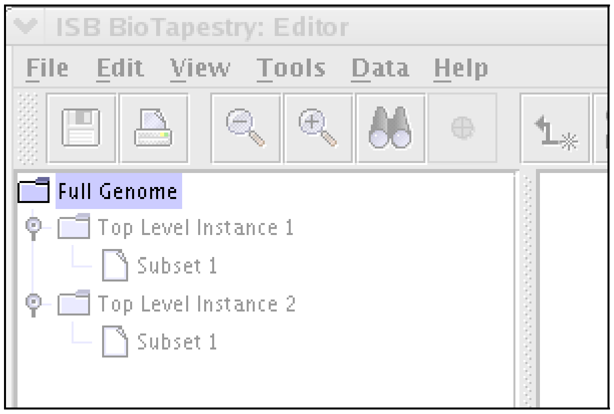
Specify Correspondence for Region PMC:

Region	Use for Model
Region P	Any Dynamic Model

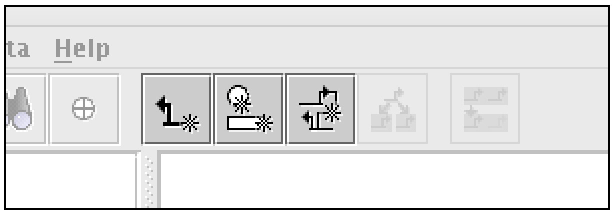
Drawing the Top Level Root Model

The network model hierarchy is created by first drawing the network at the root level. Submodels are then constructed by pushing or pulling the network elements down the hierarchy.

To draw the network, you first need to make sure that the root model is selected in the tree overview:

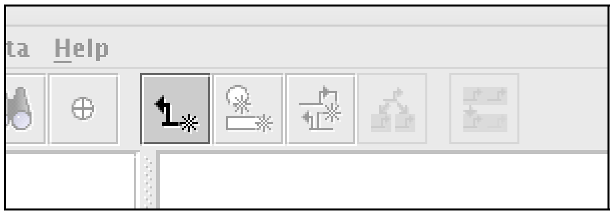


When this level is selected, the toolbar buttons to draw genes, nodes, and links are enabled:

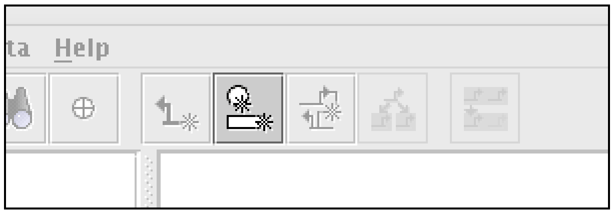


Drawing Genes and Nodes

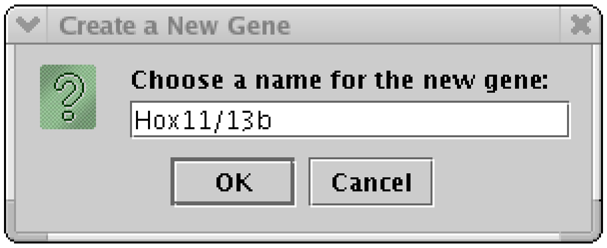
To draw a gene, click on the **Add Gene** button in the toolbar:



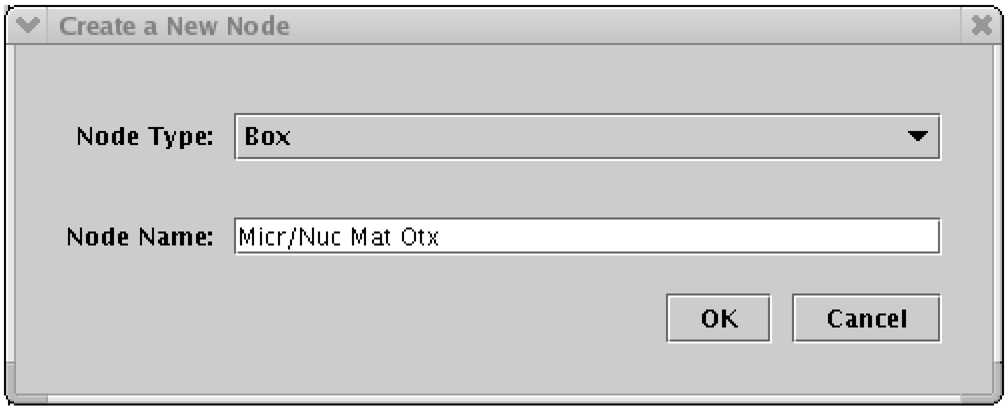
Similarly, click on the **Add Other Node** button in the toolbar to draw other types of nodes (e.g. bubbles, boxes, diamonds, etc.):



A dialog box will pop up. For genes, this box allows you to specify the gene name:



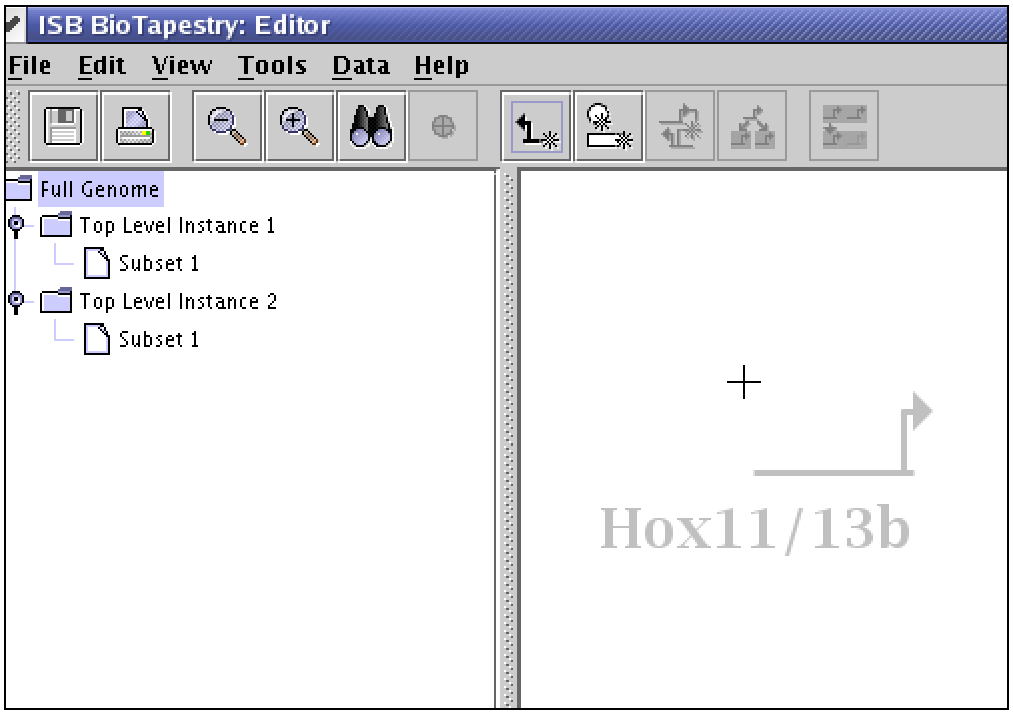
A gene name must be unique across all genes, and it cannot be empty. If you are drawing another type of node, a different dialog box appears, letting you specify the node name. A node can have any name, including blank names or duplicate names (as long as they do not share a duplicate name with a gene). This dialog box also is where you specify the visual presentation of the node: plain text, boxed text, a bubble, a diamond, an intercellular dual chevron, or a double slash representing a shorthand for an omitted process chain:


Create a New Node

Node Type: Box

Node Name: Micr/Nuc Mat Otx

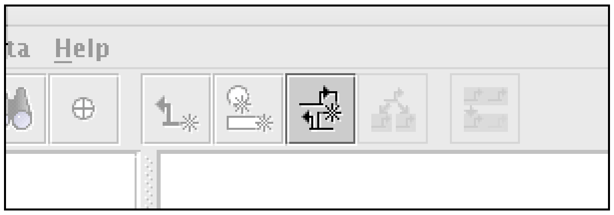
Once you have hit OK on the dialog box, the cursor changes to a cross-hair, and an image of the gene or node being placed follows the cursor. Click the mouse to place the new object. If you wish to cancel the node placement at this point, hit the **Escape** key or the **Cancel Current Add** stop sign toolbar button (not shown below).



Repeat the above process as needed to place all genes and nodes in your model.

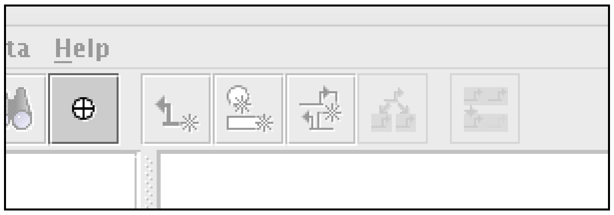
Drawing Links

To draw a link, you first click on the **Add Link** toolbar button:



This brings up a dialog box that allows you to specify the type of link and an optional link label. Once these are selected, you **click** (dragging the mouse has no effect) on a source node pad **or a corner point** of an existing link, continue clicking to create intermediate corners, and finally click on a target node link pad. (**NOTE:** Holding down the shift key during link-building clicks will constrain the link segments to be at right angles.) If you wish to cancel the link drawing before you are done, hit the **Escape** key or the **Cancel Current Add** stop sign toolbar button.

While drawing the link out, each mouse click creates a **corner point** in the link. Even if you have what appears to be a single straight link segment, it may have corner points. You can see these corner points when you select the **Toggle Link Pads** toolbar button:



When drawing a new link off an existing link, you can only start the link at an existing corner point. This may require you to add a corner point to the link. To do this, you right-click on the link where you want to add the point, and choose **Add Corner Point** from the menu.

Corner points can also be deleted by right-clicking on them. (Note that points at the intersection of two or more segments in a link tree cannot be deleted.) Also, note that certain link move operations may create two or more corner points directly on top of each other.

If you want the program to handle laying out links orthogonally for you, you can just draw direct, diagonal links between two nodes. Then you can right-click on the link and select **Auto Layout Links Through This Segment** from the menu. (Note to Mac users: If you're using a one-button mouse, press the Control key while clicking to do the same thing as right-clicking the mouse.)

Link Restrictions

There are restrictions on where you can draw links from and to. If you try to attach a link incorrectly, you should hear a beep tone. Here are the rules that are enforced:

Link corner points cannot be drawn on top of previously drawn corner points in the new link.

Links must be drawn from the source to the target, and not the reverse.

Links must originate from valid source pads. Some symbols, like genes and intercellular symbols, have only one valid source pad. With other node symbols, a pad can be either a source or a target pad.

Even on nodes where there are multiple possible source pads, only one of them can be used as a source. Instead, to draw multiple links from a single source, you need to have any additional links originate from corner points from the existing link tree from that source.

To draw a new link off an existing link tree, you need to start drawing at an existing corner point. Create a new corner point first.

Links must be drawn from the source to the target, and not the reverse.

Links must terminate at a valid target pad on a valid target node.

Links **cannot** end on a pad that is the source for another link.

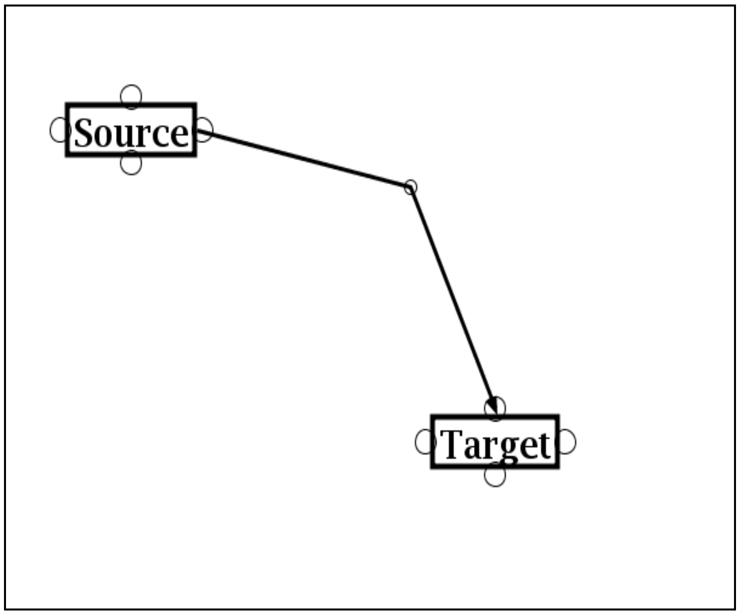
Links **can** end on a pad that is the target for another link.

Extremely short links that start and end on the same node are not allowed.

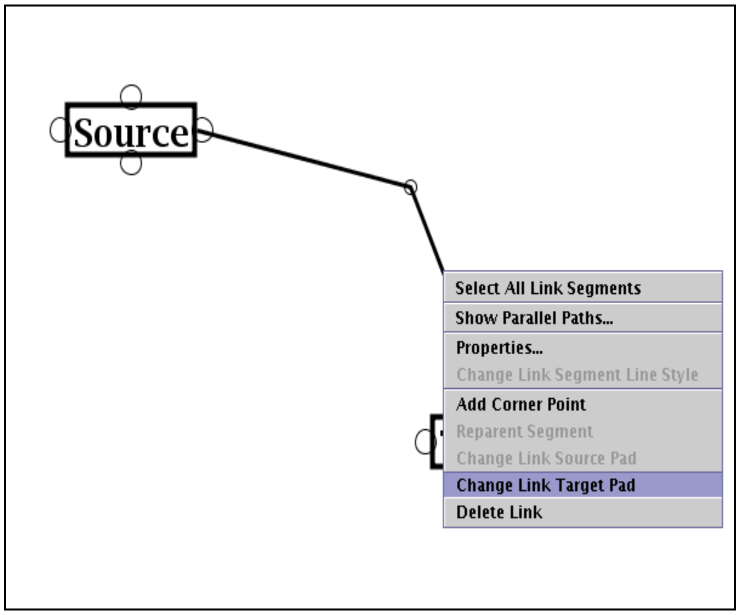
Editing Links

Once a link has been drawn, you can modify the appearance and routing of the link. In fact, though you can only draw links in the top level root model (all other levels inherit the network), the appearance and routing of a link in the top level instance model can subsequently be modified independently of the link's representation in the root model.

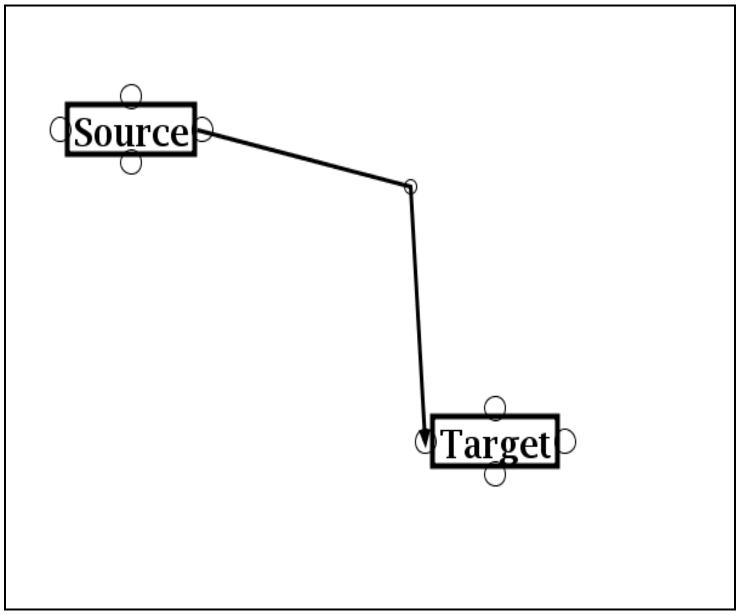
While you can change the routing of a link, you **cannot** currently change the source or destination **node** of an existing link; you can only change the launch or landing **pads** on the same node. Additionally, with link trees, you can "reparent" a link segment to another corner point. The following examples demonstrate these operations. Starting with the following simple model:



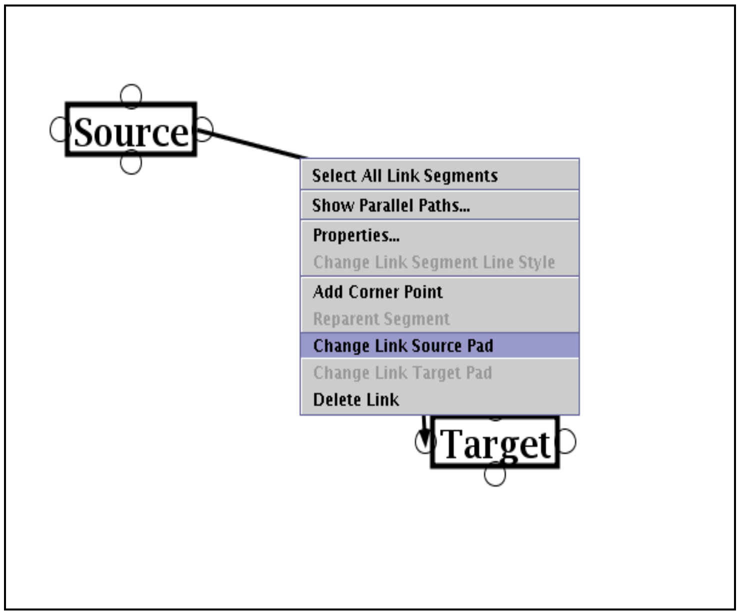
We want to change the "target pad" where the link impinges on the target node. To do this, you right click on the **terminal** segment of the link, and select **Change Link Target Pad** from the pop-up menu:



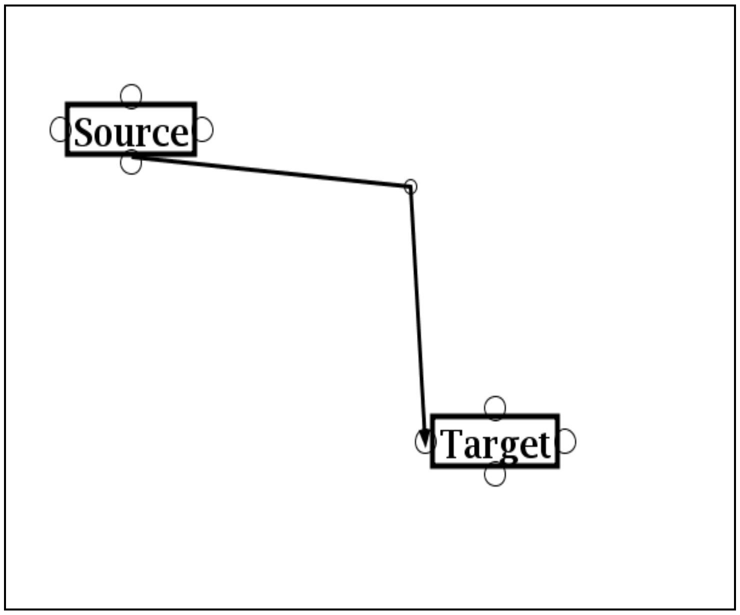
A cross-hair cursor appears, and you then click on new pad to move the link to, resulting in the following model:



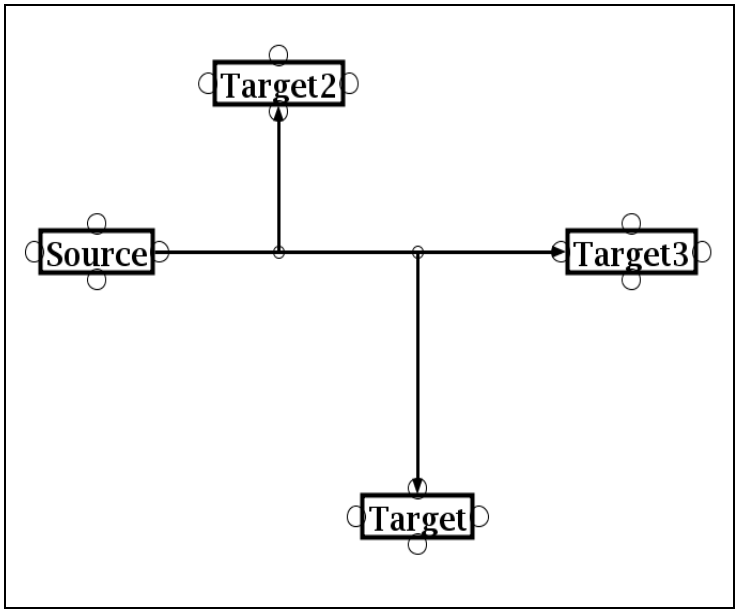
A similar operation moves the first link segment to a different source pad. Right click on the **first** segment of the link, and select **Change Link Source Pad** from the pop-up menu:



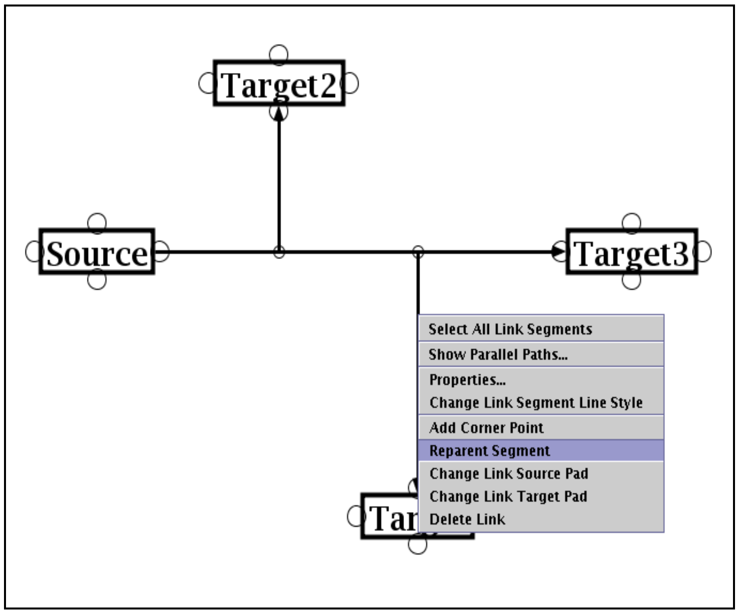
Once the cross-hair cursor appears, you can click on a new **unoccupied** pad to move the link to, resulting in the following model:



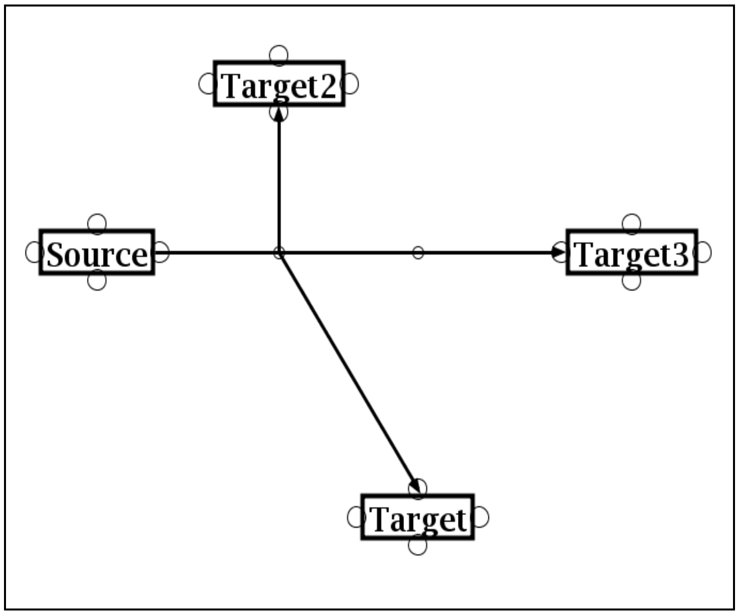
With link trees, you can "reparent" a link subtree to a new parent corner point in the link tree. For example, consider the following model that uses a link tree:



To change the topology of the link tree, right click on some segment of the tree, and select **Reparent Segment** from the pop-up menu:



Once the cross-hair cursor appears, you can click on a corner point in the same link tree, and the origin of the link segment changes to that new point. One important restriction to remember is that the new corner point you select cannot be a descendant of the link segment you are trying to move; that would disconnect a portion of the link tree from the source. Once the operation is completed, you now have the following model:

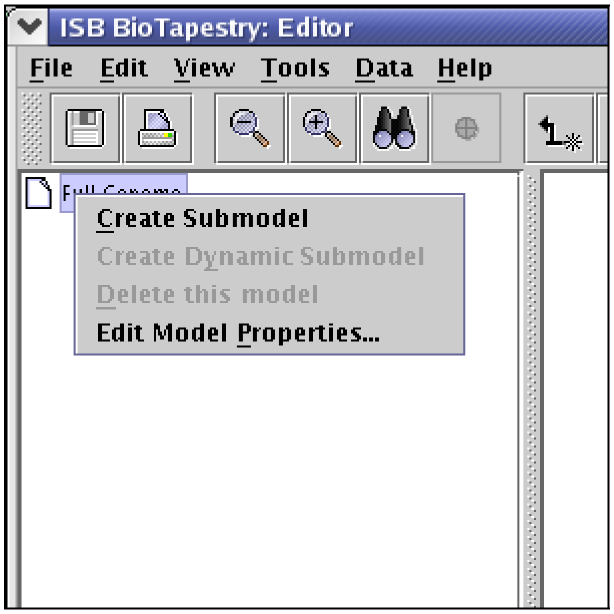


Creating Submodel Instances

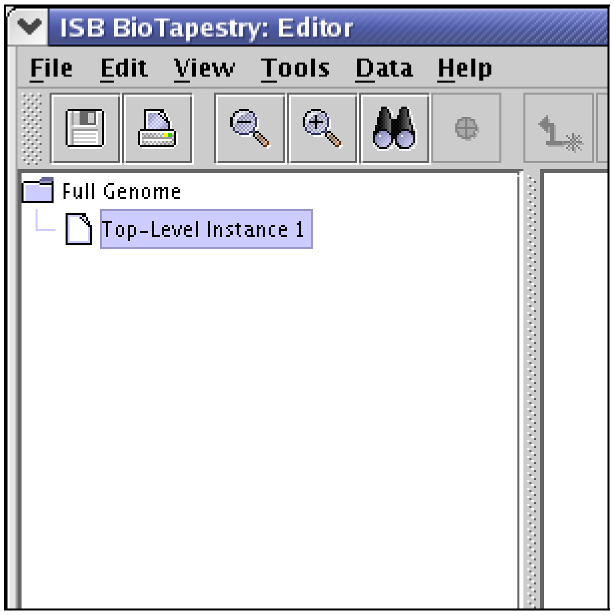
When BioTapestry starts up, it already has an empty default root model defined. While everything in the root model can be deleted, the root model itself cannot be deleted.

If you do not have any need to create submodels, all network building can occur at this root level. However, if you want to build a model hierarchy, you will need to create submodels. As the following example shows, submodels are created by right-clicking on model names in the navigation tree found at the left of the BioTapestry display.

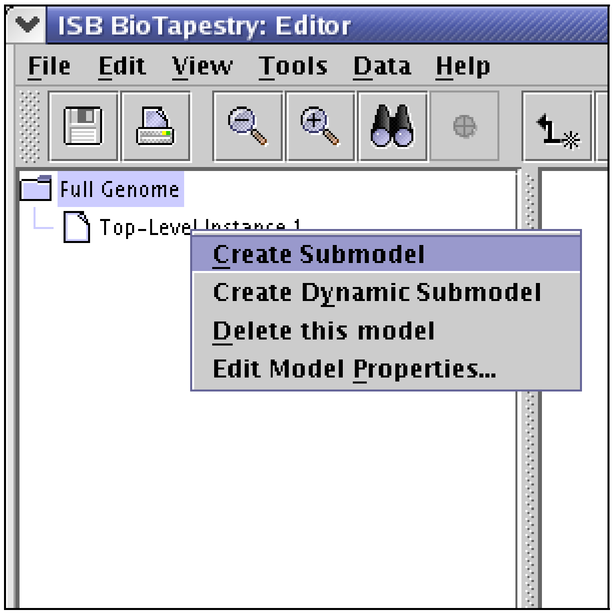
To create a submodel of the root model, right mouse click the root model, named **Full Genome**, in the navigation tree. This brings up a menu that allows you to either create a static submodel, or change the root model properties: the model's long name (which appears as a title in the model diagram), or the model description (which shows up in the description window at the bottom of the screen). Note that the **Delete this model** option is never active for the root model; you can only delete submodels. Furthermore, you cannot define a dynamic submodel of the root model. Dynamic submodels can only be created under top-level instance models.



After selecting the **Create Submodel** option, a dialog box pops up that allows you to assign a name to the new model. After you do this, the new empty top-level instance model appears under the root model:

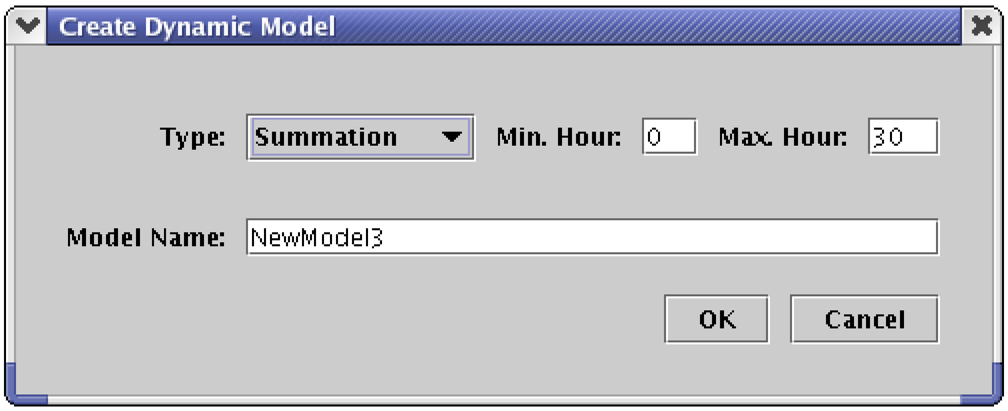


If you now right-mouse click on the top-level instance model, you get another pop-up menu:



At this point, **all** of the menu options are enabled. However, the **Create Dynamic Submodel** option will **only** be active if you have already loaded temporal expression and temporal input data into BioTapestry. This can be done using either the interactive data editing dialogs, or through XML imports.

In this example, we assume that the required experimental data has been loaded in. Select the **Create Dynamic Submodel** menu option, and the dynamic model creation dialog appears:



With this dialog, you can specify the type and time span of the new dynamic submodel. Two types of models are supported: **Summation** and **Hourly**. The summation model will display **all** the network interactions that occur within the selected time span, while hourly models provide separate hourly views of the network during the selected time span.

There are a few restrictions on building dynamic submodels:

1) The time span of a dynamic submodel must be within the time span of its parent dynamic submodel.

2) While you can have an hourly dynamic submodel beneath a summation submodel, you cannot have the reverse. Thus, hourly dynamic submodels are at the bottom of the model hierarchy.

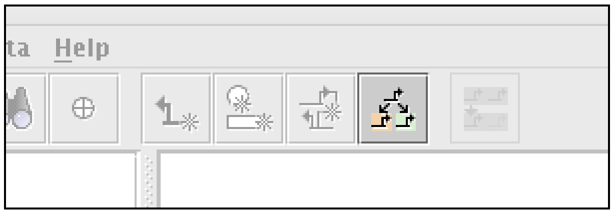
3) You cannot have a static submodel beneath a dynamic submodel.

Populating Submodels

The model hierarchy is built by either **pushing down** network elements from the root model into the top level instance models, or by **pulling down** elements from a parent model into a submodel.

Propagating from Root Model

To include items from the root model into a top-level instance, you use the **Propagate to Submodel** button in the toolbar:



Three conditions MUST be satisfied before this button is activated in the toolbar:

1) The root model must have at least one top-level instance model defined as a submodel, otherwise there is no model to push to.

2) The root model level must be selected in the tree, since we are going to be selecting items from the root model to push down.

3) There must be at least one gene, node, or link selected in the model, otherwise there is no network element to push down.

Once the three above conditions are met, the **Propagate to Submodel** button is enabled and can be pressed.

Specifying Regions

Once you click on the **Propagate to Submodel** button, if there is more than one top-level instance model, a dialog box asks you which instance model you want to propagate to. Once you have made that choice, you are asked which region you are propagating the selected network elements to. The exact sequence of dialogs then depends on the regions and their contents in the target instance model:

If BioTapestry decides that the selected items must be placed in a new region, it just asks for the name of the new region. Then, if there is already an existing region, a dialog pops up that allows you to locate the new region in the model.

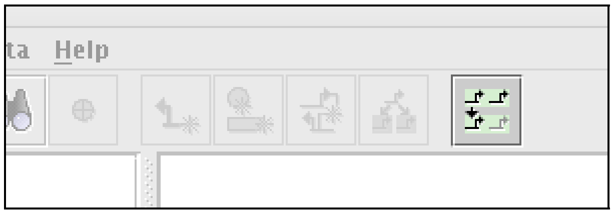
If the selected items can be placed in more than one region, BioTapestry asks which region to use. You can also create and place a new region if you wish.

If the selected items are links, BioTapestry will ask you to specify the source and target regions for the links, unless there is only one unambiguous choice. If more than one link is selected, you will only be given choices that are appropriate for the group of links as a whole. If that intersection of appropriate choices is empty, the propagation will fail.

When network elements are propagated, BioTapestry does its best to come up with a reasonable derivation of the root layout in the submodel, but since elements in the submodel layout can be moved independently of the root, the submodel layout will probably need to be modified. Also, links that bridge different regions are likely to need to be cleaned up.

Populating Subsets

Whenever a subset model is selected in the hierarchy, the **Choose Subset of Parent** button is enabled:



Once this button is pressed, the cursor changes to a crosshair, the view changes to only show the network elements present in the immediate parent, and these "ghosted" network elements can be pulled into the subset just by clicking on them. Since pressing the button toggles it on, and you can continue to pull elements in until you either click the button again, click the **Cancel Current Add** stop sign toolbar button, or hit the **Escape** key.

Static Subsets

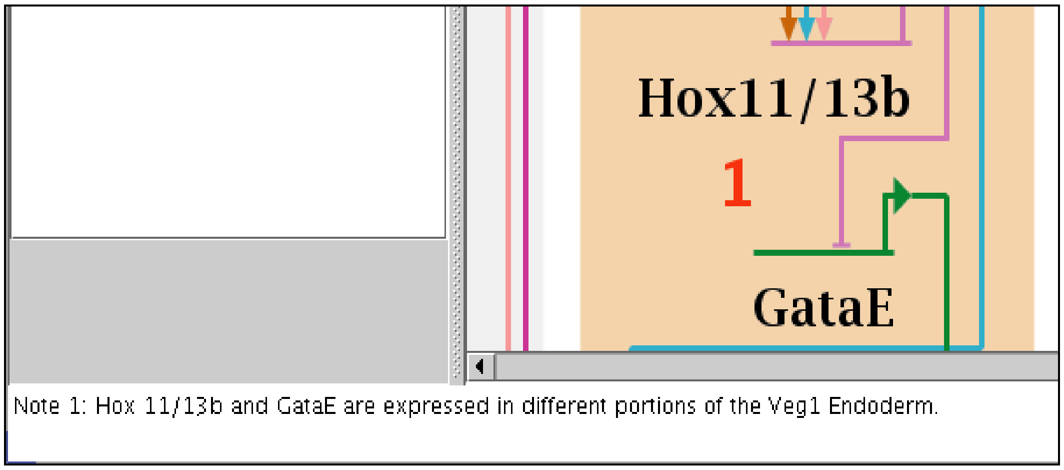
In static subsets, all elements in the parent model can be pulled down. You first need to click on the regions you wish to include, and then you can select the elements in that region. If you click on a link that does not have either the target or source already present, you are given the chance to pull them down at the same time (you must have both present to have the link).

Dynamic Subsets

In dynamic subsets, since the included links and nodes are determined by the data tables, in general you only get to choose regions to pull down into the model. From the main menu, you can also select **Edit->Add Extra Dynamic Submodel Node** to pull down extra nodes that are not part of the main regions you have included in the dynamic model. This is useful for including boundary interconnect nodes belonging to a neighboring region without having to include the entire region.

Adding Notes

Notes can be added to any model except the root model. They are unique in that the user can define different notes for each submodel; unlike other network components, they are **not** shared in a hierarchical fashion. To add a note, select **Edit->Add Note...** from the main menu, provide the note name and text, and then click to place the note in the model. Then, when the user clicks on the note, the note contents are displayed in the BioTapestry description area:



Note 1: Hox 11/13b and GataE are expressed in different portions of the Veg1 Endoderm.

Right-clicking on the note allows you to edit or delete it.

The Experimental Data

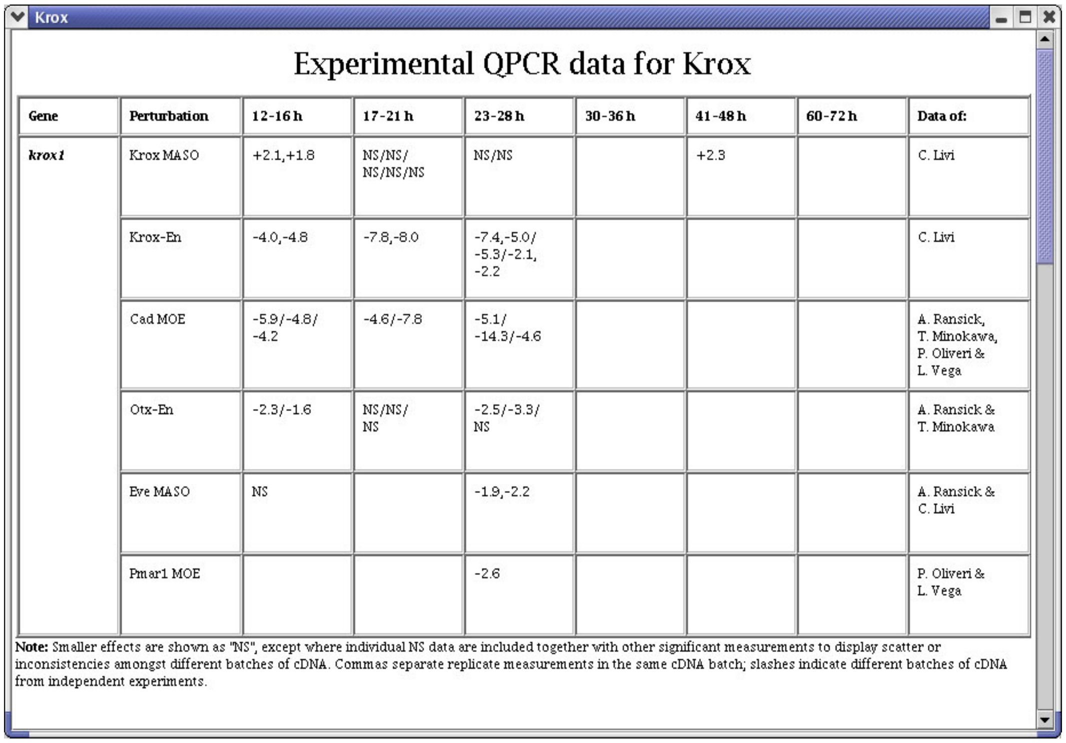
There are three roles for experimental data in BioTapestry:

1) Data can be associated with the elements of a previously constructed network drawing. These data can be accessed for display by right-clicking on the network elements.

2) The time expression and temporal input data tables can be used to populate dynamic submodels.

3) Current development of BioTapestry is focusing on how to take the raw QPCR and time expression data and use to help the researcher construct the model hierarchy.

QPCR Data



Experimental QPCR data for Krox

Gene	Perturbation	12-16 h	17-21 h	23-28 h	30-36 h	41-48 h	60-72 h	Data of:
krox1	Krox MASO	+2.1,+1.8	NS/NS/ NS/NS/NS	NS/NS		+2.3		C. Livi
	Krox-En	-4.0,-4.8	-7.8,-8.0	-7.4,-5.0/ -5.3/-2.1, -2.2				C. Livi
	Cad MOE	-5.9/-4.8/ -4.2	-4.6/-7.8	-5.1/ -14.3/-4.6				A. Ransick, T. Minokawa, P. Oliveri & L. Vega
	Otx-En	-2.3/-1.6	NS/NS/ NS	-2.5/-3.3/ NS				A. Ransick & T. Minokawa
	Eve MASO	NS		-1.9,-2.2				A. Ransick & C. Livi
	Pmar1 MOE			-2.6				P. Oliveri & L. Vega

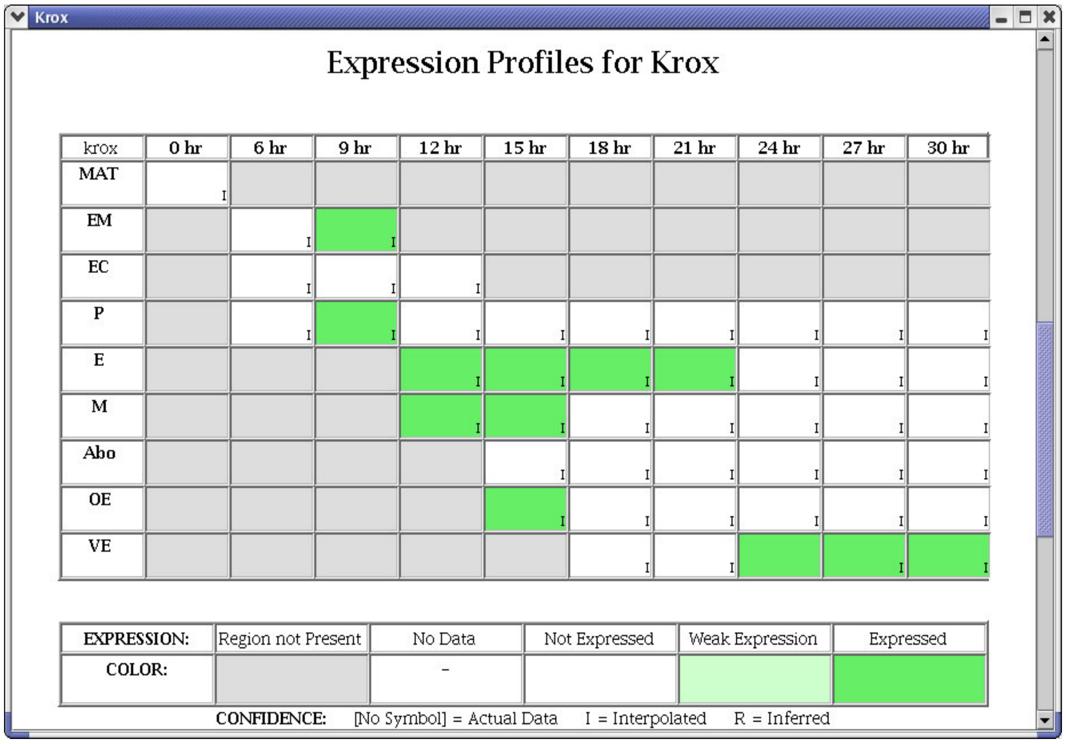
Note: Smaller effects are shown as "NS", except where individual NS data are included together with other significant measurements to display scatter or inconsistencies amongst different batches of cDNA. Commas separate replicate measurements in the same cDNA batch; slashes indicate different batches of cDNA from independent experiments.

BioTapestry provides a framework for managing and displaying the quantitative polymerase chain reaction (QPCR) perturbation data used to derive the interconnectivity of the network model. The QPCR data is organized into a separate table for each target gene, where each row of the table contains a different perturbation. Currently, the perturbations type supported are morpholino-substituted antisense oligo nucleotides (MASO), messenger RNA overexpression (MOE) and engrailed repressor domain (EN). For each perturbation, data is organized into columns for different time spans, and within each span, different measurements are organized into batches, where commas separate replicate measurements in the same cDNA batch, and slashes indicate different batches of cDNA from independent experiments.

In addition, the QPCR data management framework includes a **Null Perturbations** table, which summarizes those genes that were unaffected by each perturbation experiments. Finally, the framework provides a footnote table, where the annotations to the data entries in the main table are managed.

QPCR data are not used to drive the dynamic data displays; that role is filled by the temporal input data sets described below. However, it is currently possible to construct the root model from the QPCR data, and there is work in progress to provide analysis tools to assist the researcher in reducing this raw interaction data to the temporal input data sets.

Time Expression Data



Expression Profiles for Krox

krox	0 hr	6 hr	9 hr	12 hr	15 hr	18 hr	21 hr	24 hr	27 hr	30 hr
MAT	I	▨	▨	▨	▨	▨	▨	▨	▨	▨
EM	▨	I	🟩 I	▨	▨	▨	▨	▨	▨	▨
EC	▨	I	I	I	▨	▨	▨	▨	▨	▨
P	▨	I	🟩 I	I	I	I	I	I	I	I
E	▨	▨	▨	🟩 I	🟩 I	🟩 I	🟩 I	I	I	I
M	▨	▨	▨	🟩 I	🟩 I	I	I	I	I	I
Abo	▨	▨	▨	▨	I	I	I	I	I	I
OE	▨	▨	▨	▨	🟩 I	I	I	I	I	I
VE	▨	▨	▨	▨	▨	I	I	🟩	🟩 I	🟩 I

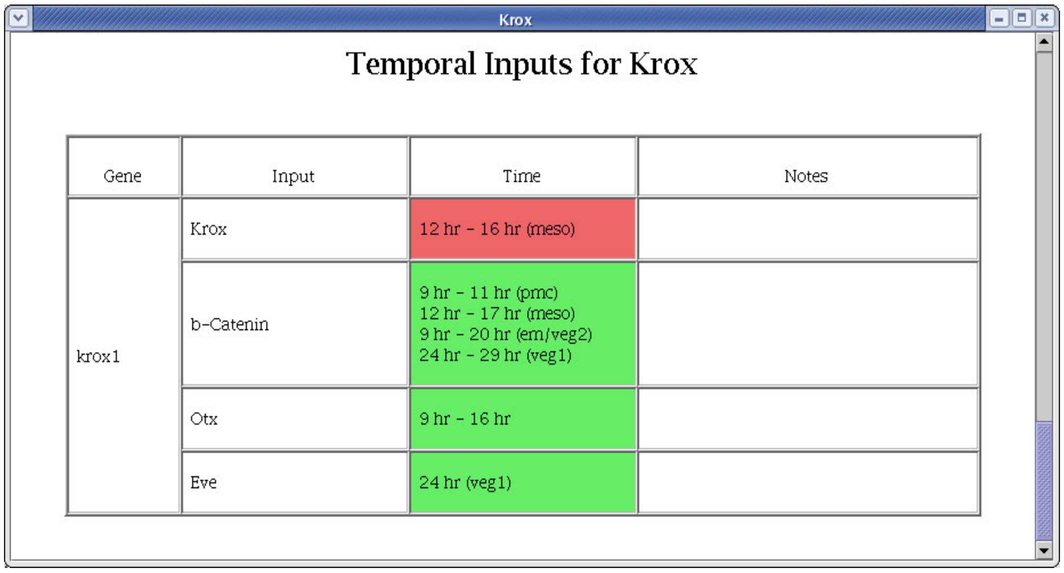
EXPRESSION:	Region not Present	No Data	Not Expressed	Weak Expression	Expressed
COLOR:	▨	-		░	🟩

CONFIDENCE: [No Symbol] = Actual Data I = Interpolated R = Inferred

Time expression data describes the times and the regions where each gene is being expressed. Currently, the framework provides a discrete set of four possible categories for the expression: **No Data**, **No Expression**, **Weak Expression**, and **Expressed**. In addition, each data entry can be tagged with a attribute describing the confidence level (e.g. normal, interpolated, inferred, etc.).

Time expression data is used to decide when genes and other network nodes should be shown active or inactive in a dynamic model display. When used for this purpose, **No Data** and **No Expression** entries are depicted as "off", and **Weak Expression** and **Expressed** are depicted as "on".

Temporal Input Data



Temporal Inputs for Krox

Gene	Input	Time	Notes
krox1	Krox	12 hr - 16 hr (meso)	
	b-Catenin	9 hr - 11 hr (pmc) 12 hr - 17 hr (meso) 9 hr - 20 hr (em/veg2) 24 hr - 29 hr (veg1)	
	Otx	9 hr - 16 hr	
	Eve	24 hr (veg1)	

Temporal input data describes the inputs to a gene and the times when each input is active. These data entries are used to decide when links in the network are shown as active or inactive in a dynamic model display.

Note that the time span when a given input to a gene is active does not have to correspond exactly to the time expression of that gene. In fact, it is frequently the case that a promoter link is shown as active before the gene is shown active, or a repressor link is active when the gene is not.

Frequently, a temporal input entry is tagged with a particular region, since the start time and length of an input to a gene are often different in different regions. However, a region does not have to be specified if the meaning is unambiguous. For example, the behavior could be identical across all regions, or the gene may only be depicted in one region of the network during the time the input is active.

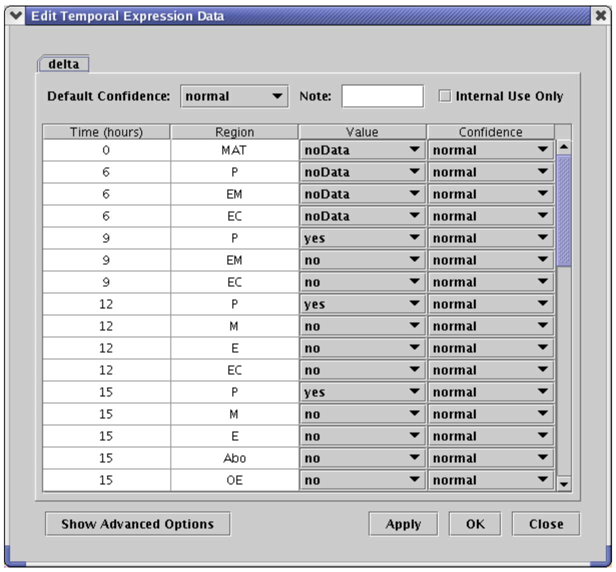
Mapping Between Data and the Models

It is critical to understand that the data in the data tables needs to be connected to the elements in the network model. As long as the name of the gene or node in the network matches the name of the experimental data entry (e.g. gene **Pmar1** and the data entries tagged **pmar1**), BioTapestry assumes that the two are connected. (This name match is case and space-insensitive, e.g. **Pmar 1** and **pmar1** will match.) Note that regions also need to be associated with region references in the data tables.

However, sometimes more control over this association is needed. For example, you may have three different QPCR data entries, e.g. **fvmo1**, **fvmo2**, and **fvmo3**, but only one gene in the network that is used as a shorthand representation for all three, e.g. **FvMo1,2,3**. Or you may have a node that is represented with an abbreviated name in the network, e.g. **unkn pmc rep**, but it is shown in the data table as **Unknown Micromere Repressor**. In these cases, you need to explicitly create an association between the network element and the data.

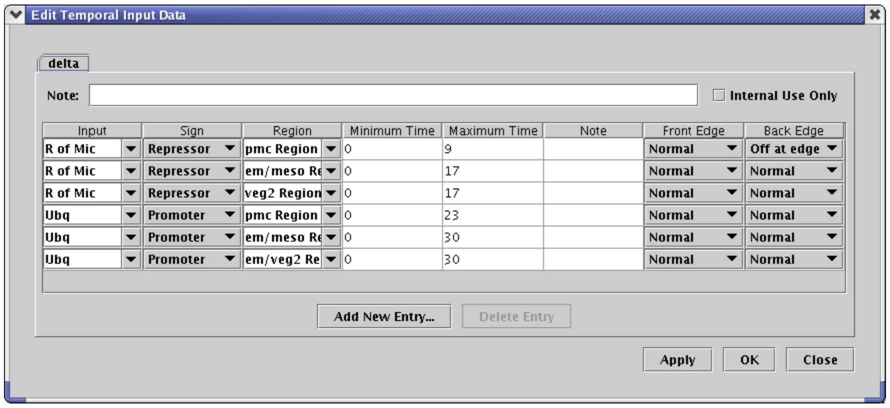
Setting Up Dynamic Models

Before the genes and links in a dynamic submodel can be included automatically, certain preliminary steps need to be carried out. Of course, the time expression data (which specifies at what times and in what regions that each gene is active) needs to be entered, as does the temporal input data (which describes the inputs to a gene and the times when each input is active). This information can be imported using XML files, or it can be entered using interactive dialogs:



Default Confidence: normal
Note:
☐ Internal Use Only

Time (hours)	Region	Value	Confidence
0	MAT	noData	normal
6	P	noData	normal
6	EM	noData	normal
6	EC	noData	normal
9	P	yes	normal
9	EM	no	normal
9	EC	no	normal
12	P	yes	normal
12	M	no	normal
12	E	no	normal
12	EC	no	normal
15	P	yes	normal
15	M	no	normal
15	E	no	normal
15	Abo	no	normal
15	OE	no	normal



Edit Temporal Input Data

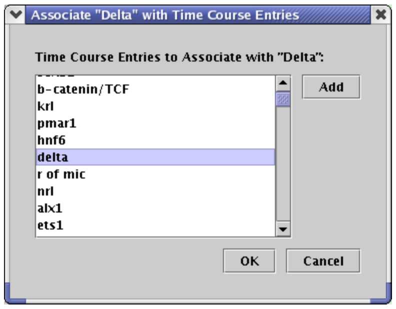
delta

Note:

Internal Use Only

Input	Sign	Region	Minimum Time	Maximum Time	Note	Front Edge	Back Edge
R of Mic	Repressor	pmc Region	0	9		Normal	Off at edge
R of Mic	Repressor	em/meso Re	0	17		Normal	Normal
R of Mic	Repressor	veg2 Region	0	17		Normal	Normal
Ubq	Promoter	pmc Region	0	23		Normal	Normal
Ubq	Promoter	em/meso Re	0	30		Normal	Normal
Ubq	Promoter	em/veg2 Re	0	30		Normal	Normal

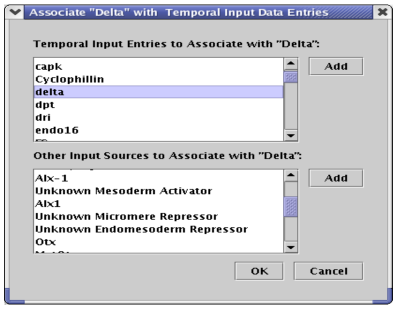
If the experimental data entries have names that match the names used in the network model (e.g. data entries labeled **pmar1** match the gene labeled **Pmar1** in the network diagram, and the region tag **PMC** used in the data tables matches the region with the same name in the network diagram), BioTapestry can link the data to the network without any extra steps. However, if you wish to tie a gene labeled **FvMo1,2,3** to the three separate data table entries **fvmo1**, **fvmo2**, and **fvmo3**, you will need to set up a custom data association:



Associate "Delta" with Time Course Entries

Time Course Entries to Associate with "Delta":

b-catenin/TCF
krl
pmar1
hnf6
delta
r of mic
nrl
alx1
ets1



Associate "Delta" with Temporal Input Data Entries

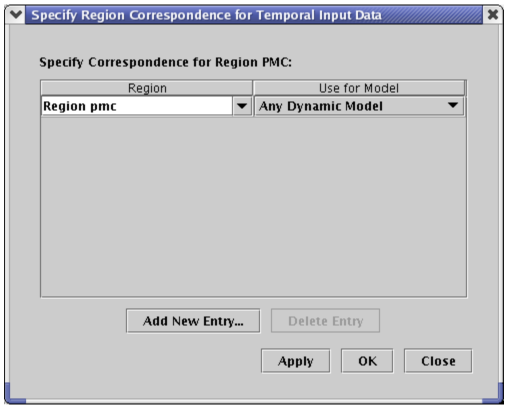
Temporal Input Entries to Associate with "Delta":

capk
Cyclophillin
delta
dpt
dri
endo16

Other Input Sources to Associate with "Delta":

Alx-1
Unknown Mesoderm Activator
Alx1
Unknown Micromere Repressor
Unknown Endomesoderm Repressor
Otx

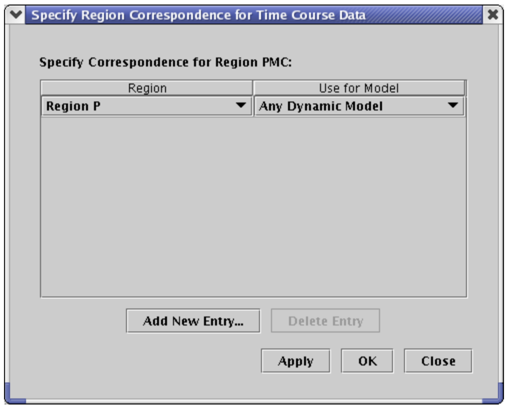
Regions follow the same naming rules, and may also require the user to set up a custom association (e.g. to link the region named **PMC** with data table entries labeled simply **P**:



Specify Region Correspondence for Temporal Input Data

Specify Correspondence for Region PMC:

Region	Use for Model
Region pmc	Any Dynamic Model



Specify Region Correspondence for Time Course Data

Specify Correspondence for Region PMC:

Region	Use for Model
Region P	Any Dynamic Model

Building the Experimental Datasets

The QPCR, temporal expression, and temporal input experimental data can be entered into BioTapestry in one of three ways:

1) The data can be imported from external files that are in a specific XML format used internally by BioTapestry. **WARNING:** This import operation is currently not checking the imported data for internal consistency. It is the user's responsibility to insure data correctness. Inconsistent data can produce unexpected results, including crashing the program.

2) QPCR data records can be read in using a specific comma-separated value (CSV) format. This is a good way to keep the program updated with ongoing QPCR experimental results.

3) The user can enter the data interactively using the editing dialogs provided by BioTapestry.

Importing Experimental Data

If the QPCR, temporal expression, or temporal input data are formatted using BioTapestry's XML input format, large amounts of data can be imported directly into BioTapestry:

```
<targetGene name="sm30">
  <perturbations>
    <perturbation>
      <sources>
        <source type="MASO">Dri</source>
      </sources>
      <investigators>
        <name>G. Amore</name>
      </investigators>
      <times>
        <timeSpan span="17-21 h" >
          <batch>
            <measurement value="-6.2" />
          </batch>
          <batch>
            <measurement value="-5.4" />
          </batch>
        </timeSpan>
      </times>>
    </perturbation>
  </perturbations>
</targetGene>
```

An alternative pathway to import QPCR data is using a comma-separated value format:

```
"FoxA","MASO"
"48h","1/29/2004"
"Rachel Gray"
"FoxA","2.57","GataE","-0.01","Endo16","0.39"

"GataE","MASO"
"24h","1/26/2004"
"Rachel Gray"
"Apobec","-9.12","FoxB","-4.87","Brac","-4.15"
```

Entering Experimental Data

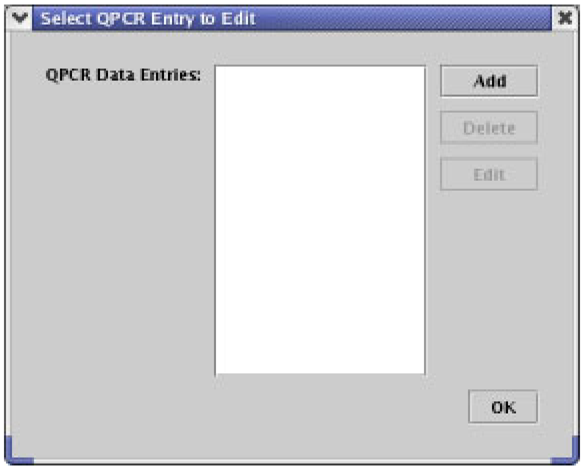
There is an extensive set of dialogs available in BioTapestry for directly entering experimental data for QPCR, time expression, and temporal inputs.

QPCR Data

You can access the QPCR data tables by two different routes. Even if you have no network drawn, you can create and edit QPCR tables by selecting **Data->QPCR Data** from the main menu. You can always access the footnote and null perturbation tables, but you need to define the time spans of the columns before you can work with the main QPCR data. If you try to edit a QPCR entry without having defined the columns, BioTapestry will present you with the column setup dialog before you can proceed.

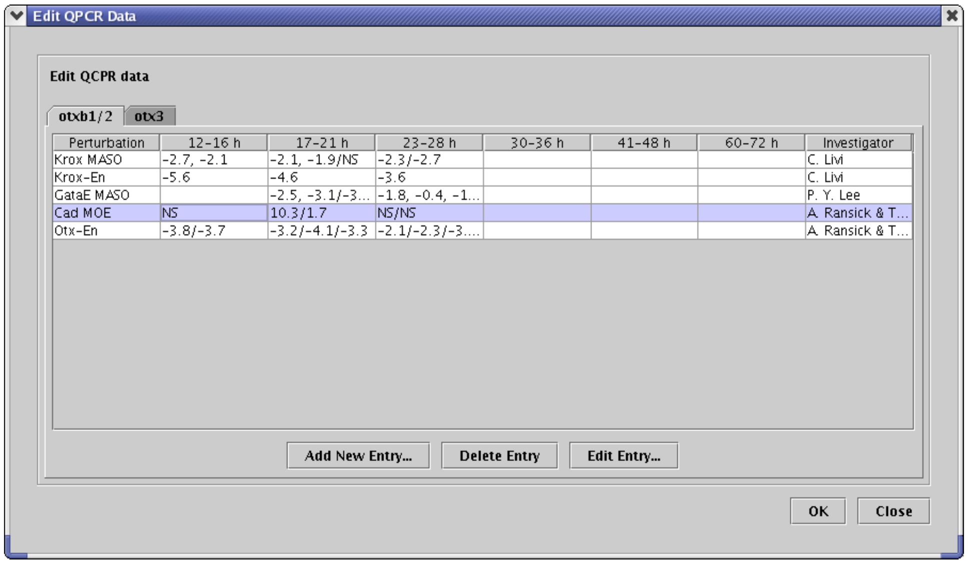
The second way to access QPCR data is to right click on a gene in the network diagram. You then select **Manage QPCR Data->Edit Data...** to bring up the QPCR editing dialog.

If you go the route of selecting **Data->QPCR Data->Manage Full QPCR Data Table** from the main menu, you are presented with the list of QPCR entries to edit:



If there are no entries defined, click the **Add** button, enter a unique name for the new entry, and click OK (you may need to enter column definitions first). Once the entry is defined, select it from the list and click the **Edit** button, which brings up an **Edit QPCR Data** dialog.

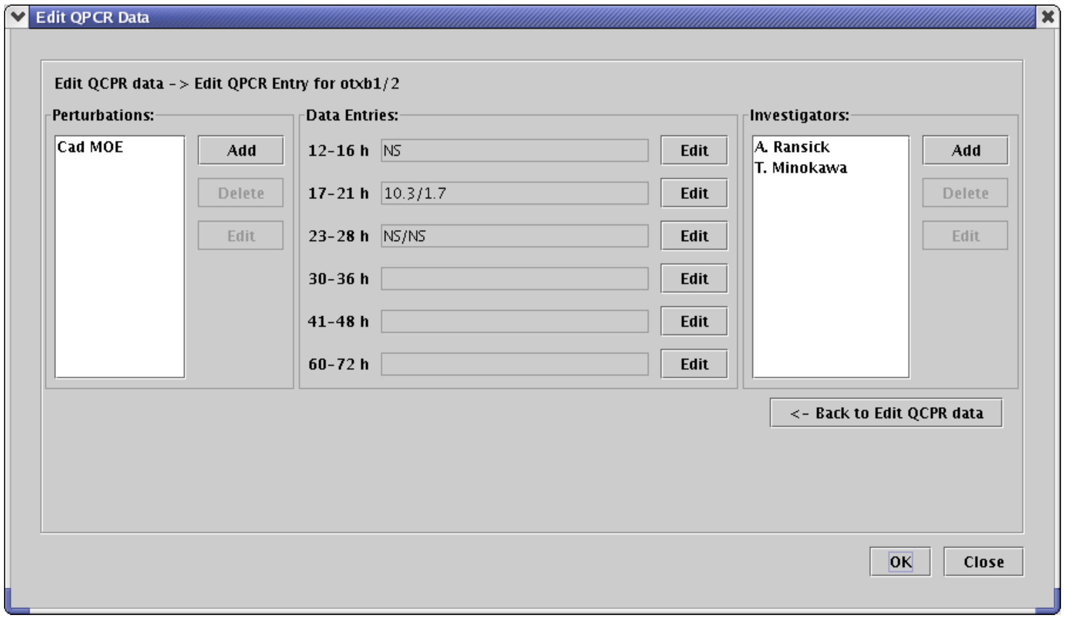
The edit dialogs that appear for the two different routes are slightly different, depending on how you got there. From the main menu route, the dialog gives you an opportunity to rename the entry, and you have only one tab for a single table. By right clicking, you cannot edit the table name, but there is a tab for each QPCR table associated with the gene you clicked on. The following figure shows the right-click dialog:



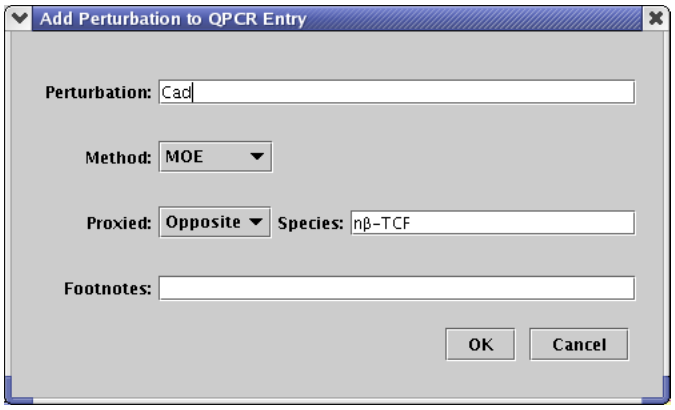
Perturbation	12-16 h	17-21 h	23-28 h	30-36 h	41-48 h	60-72 h	Investigator
Krox MASO	-2.7, -2.1	-2.1, -1.9/NS	-2.3/-2.7				C. Livi
Krox-En	-5.6	-4.6	-3.6				C. Livi
GataE MASO		-2.5, -3.1/-3...	-1.8, -0.4, -1...				P. Y. Lee
Cad MOE	NS	10.3/1.7	NS/NS				A. Ransick & T...
Otx-En	-3.8/-3.7	-3.2/-4.1/-3.3	-2.1/-2.3/-3....				A. Ransick & T...

The QPCR data organization is complex. Tables contain perturbation rows, each row may contain several perturbation sources, as well as data for different time spans, and each time span contains one or more batches of data, with multiple measurements per batch. BioTapestry presents a single dialog for editing all these entries, where by clicking on an entry, the dialog view is replaced by a new view for that entry. You can either continue to drill down through the views, or back out at any time. The changes you make while navigating do not take effect until you click the **OK** button for the dialog.

In the top level QPCR editing window, you can highlight an entry by clicking on it, and then press the **Edit Entry...** button to drill down on that entry. The dialog view is then replaced by a view for that one entry:



In this view, you can add, edit, and delete perturbation sources in the left panel and investigators in the right panel. If you select a perturbation source and click the **Edit** button, a dialog box appears that allows you to set the characteristics of the source:



Add Perturbation to QPCR Entry

Perturbation: Cad

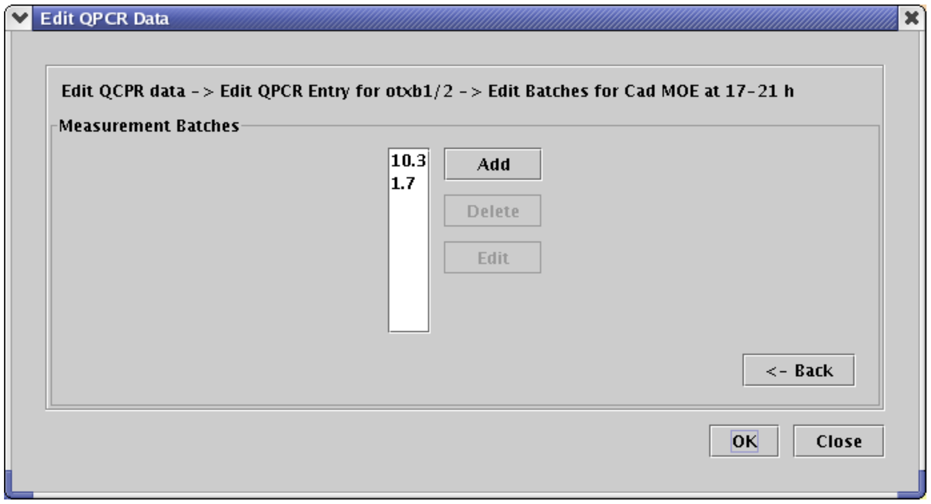
Method: MOE

Proxied: Opposite   Species: nβ-TCF

Footnotes:

The **Proxied** entry requires some explanation. Some perturbation sources (such as cadherin) are not directly depicted in the network, but influence chemical species that are in the network. In the above example, increased cadherin levels reduce beta-catenin; since the latter is represented in the diagram, we tag cadherin as a proxy for beta-catenin.

To edit a data entry, click on the **Edit** button for that entry time span, and the dialog then presents a view that allows you to manipulate separate batches:



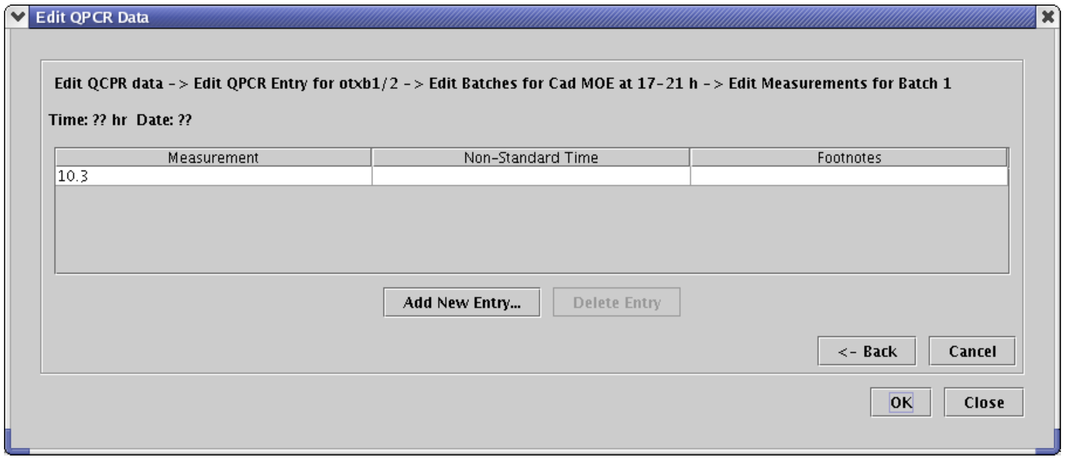
Edit QPCR Data

Edit QCPR data -> Edit QPCR Entry for otxb1/2 -> Edit Batches for Cad MOE at 17-21 h

Measurement Batches

10.3
1.7

If you then select a batch from the list and click the **Edit** button, the dialog presents a table with a row for each measurement. You can then directly edit the values in each cell of the table:

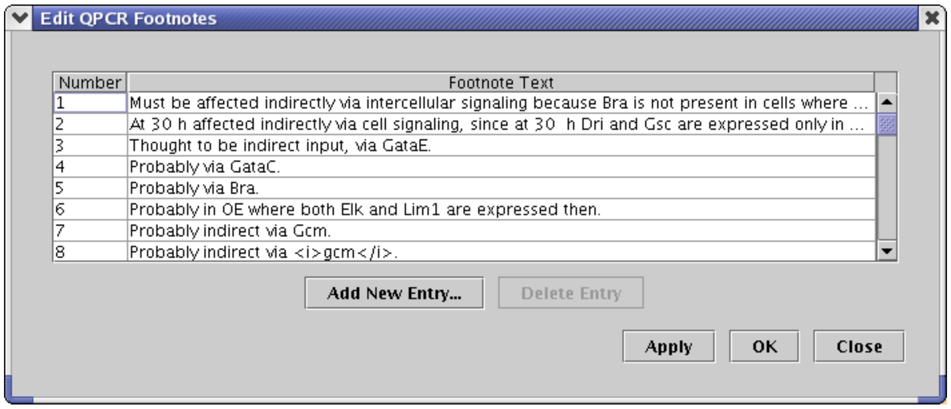


Edit QCPR data -> Edit QPCR Entry for otxb1/2 -> Edit Batches for Cad MOE at 17-21 h -> Edit Measurements for Batch 1

Time: ?? hr Date: ??

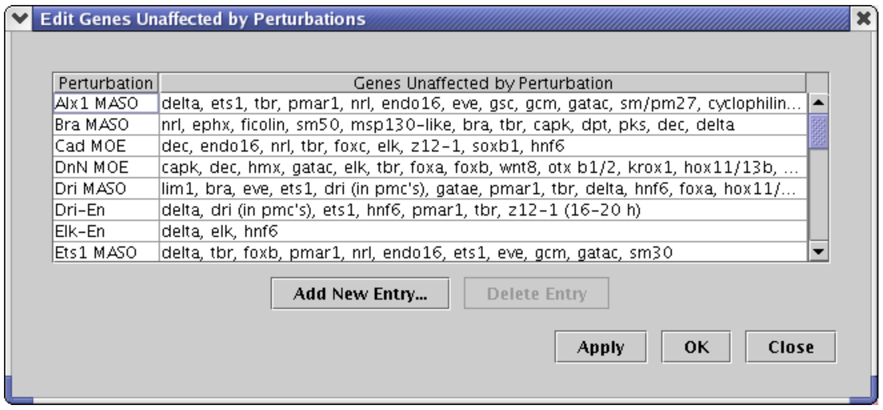
Measurement	Non-Standard Time	Footnotes
10.3		

If you want to include footnotes to annotate perturbation sources or measurements, you need to create those footnotes using the footnote table. Selecting **Data->QPCR Data->Footnotes** from the main menu brings up the Footnote table:



Number	Footnote Text
1	Must be affected indirectly via intercellular signaling because Bra is not present in cells where ...
2	At 30 h affected indirectly via cell signaling, since at 30 h Dri and Gsc are expressed only in ...
3	Thought to be indirect input, via GataE.
4	Probably via GataC.
5	Probably via Bra.
6	Probably in OE where both Elk and Lim1 are expressed then.
7	Probably indirect via Gcm.
8	Probably indirect via <i>gcm</i>.

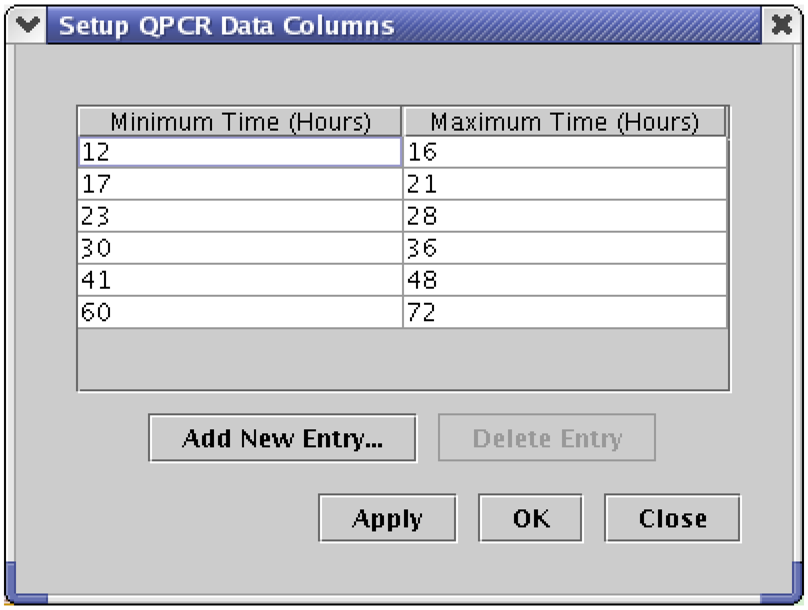
The final table in the QPCR data is the Null Perturbations table, which summarizes those genes that are unaffected each perturbation. Selecting **Data->QPCR Data->Null Perturbations** from the main menu brings up this table:



Perturbation	Genes Unaffected by Perturbation
Alx1 MASO	delta, ets1, tbr, pmar1, nrl, endo16, eve, gsc, gcm, gatac, sm/pm27, cyclophilin...
Bra MASO	nrl, ephx, ficolin, sm50, msp130-like, bra, tbr, capk, dpt, pks, dec, delta
Cad MOE	dec, endo16, nrl, tbr, foxc, elk, z12-1, soxb1, hnf6
DnN MOE	capk, dec, hmx, gatac, elk, tbr, foxa, foxb, wnt8, otx b1/2, krox1, hox11/13b, ...
Dri MASO	lim1, bra, eve, ets1, dri (in pmc's), gatae, pmar1, tbr, delta, hnf6, foxa, hox11/...
Dri-En	delta, dri (in pmc's), ets1, hnf6, pmar1, tbr, z12-1 (16-20 h)
Elk-En	delta, elk, hnf6
Ets1 MASO	delta, tbr, foxb, pmar1, nrl, endo16, ets1, eve, gcm, gatac, sm30

Setup

It was mentioned above that BioTapestry requires you to setup the QPCR data columns before entries can be created. Selecting **Data->QPCR Data->Setup QPCR Table Columns** from the main menu brings up the Setup table:

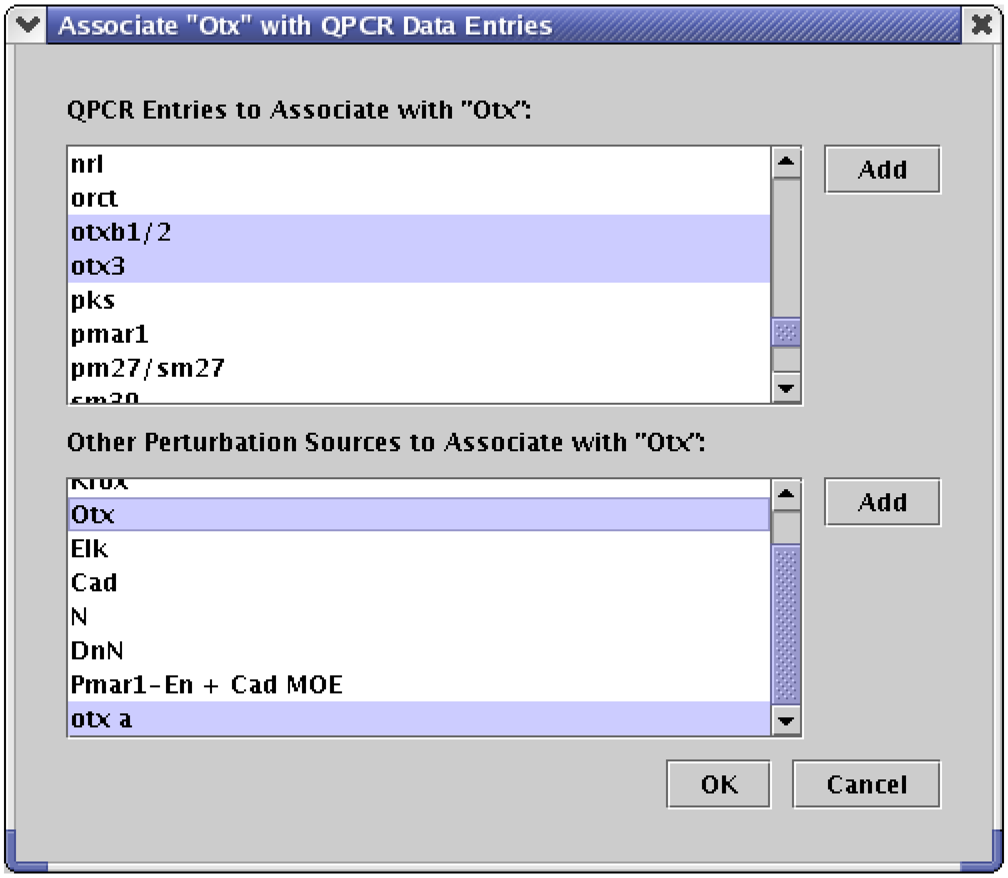


Minimum Time (Hours)	Maximum Time (Hours)
12	16
17	21
23	28
30	36
41	48
60	72

You need to specify any number of rows by entering minimum and maximum hours for each column. While the time spans must be increasing and cannot be overlapping, gaps between time spans are allowed. **NOTE:** Version 1.0 of BioTapestry does not allow you to change these column definitions once you have started entering data into the tables!

Data Mapping

Data mapping is the mechanism by which a user can click on a gene labeled **FvMo1,2,3** and have the three separate QPCR tables **fvmo1**, **fvmo2**, and **fvmo3** be displayed. For most cases where the names in the network diagram match the data tables entries (this match is case- and space-insensitive), you don't have to worry about this feature. However, to set up a custom association, you can select **Manage QPCR Data->Custom Association to Data...** from the right click menu to get the following dialog box:

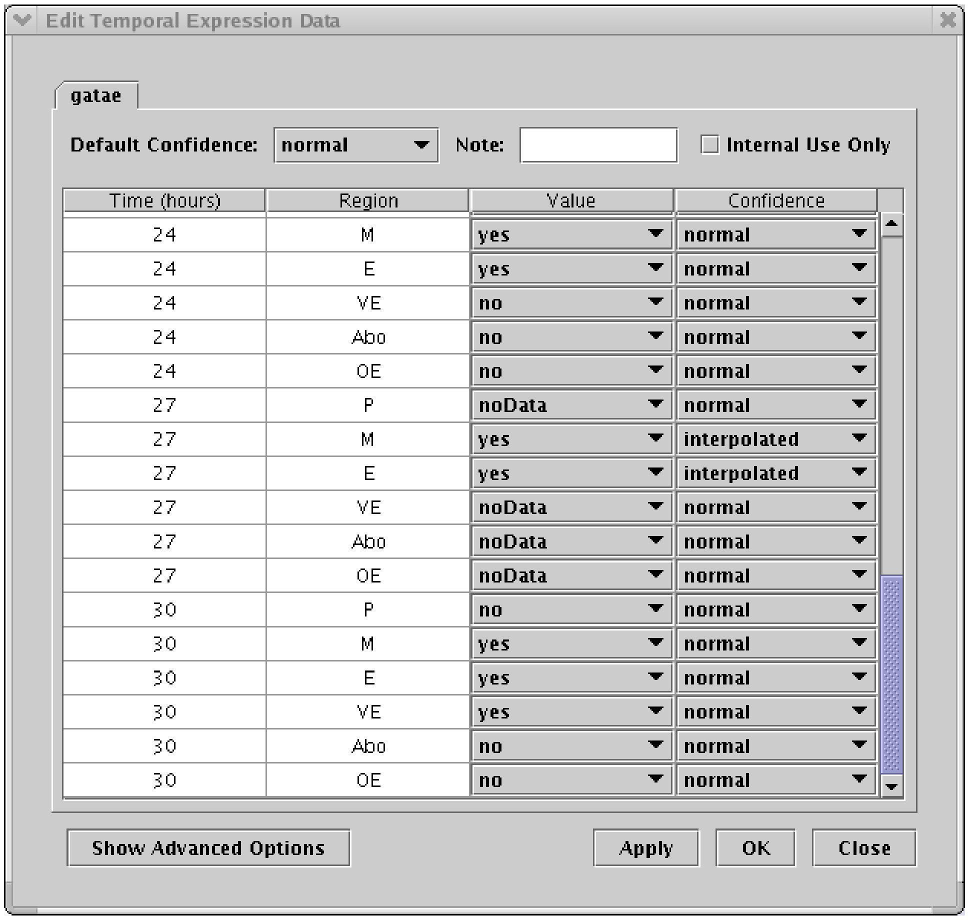

Associate "Otx" with QPCR Data Entries
QPCR Entries to Associate with "Otx":
nrl
orct
otxb1/2
otx3
pks
pmar1
pm27/sm27
Add
Other Perturbation Sources to Associate with "Otx":
Otx
Elk
Cad
N
DnN
Pmar1-En + Cad MOE
otx a
Add
OK
Cancel

For QPCR data, this dialog has two separate lists. The first list sets an association between the network gene and one or more QPCR tables. The second list associates the network gene with the perturbation rows present in one or more of all the QPCR tables.

The selections in the list indicate what QPCR elements to associate with the gene in the network, while the **Add** button is just used to create a new entry in the list. To select or unselect multiple entries in a list, hold down the **Shift** key or the **Ctrl** key while clicking on the entry.

Time Expression Data

Time expression data specifies what times and what regions that each gene is active. This information is used directly by dynamic submodels to determine is a gene is shown to be active at a given hour and region. Currently, the expression level of each gene is set to four categories: no data, no expression, weak expression, and full expression. For display purposes, weak and full expression are both shown as "on", while no expression and no data entries are shown off. The expression data can be edited using the editing dialog, which is accessed off the Main menu by selecting **Data->Time Course Data->Manage Full Time Course Data Table** and then choosing an entry to edit, or by right clicking on the gene and selecting **Manage Time Course Data->Edit Data...**:



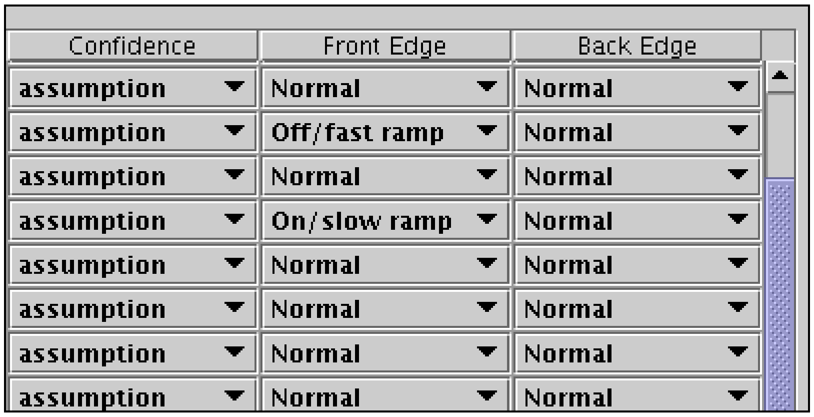
Default Confidence: normal
Note:
Internal Use Only

Time (hours)	Region	Value	Confidence
24	M	yes	normal
24	E	yes	normal
24	VE	no	normal
24	Abo	no	normal
24	OE	no	normal
27	P	noData	normal
27	M	yes	interpolated
27	E	yes	interpolated
27	VE	noData	normal
27	Abo	noData	normal
27	OE	noData	normal
30	P	no	normal
30	M	yes	normal
30	E	yes	normal
30	VE	yes	normal
30	Abo	no	normal
30	OE	no	normal

Five different confidence levels can be selected. All levels except "normal" are tagged with a letter (A, I, R, or Q) in the temporal expression data displays.

If the user selects the **Show Advanced Options** button, he can then specify the edge interpolation strategies, which modifies how the expression turns on or off at hour intervals between those given in the data tables. For example (top numbers are actual data points in 3-hour intervals):

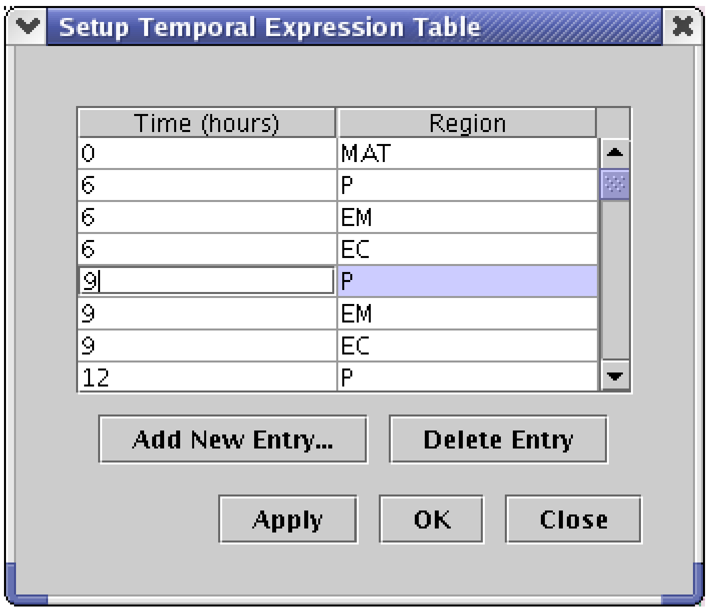
```
                     3     6     9
Front edge on slow:  0 1 1 1 1 1 1
Front edge on fast:  0 0 1 1 1 1 1
Front edge normal:   0 0 0 1 1 1 1
Front edge off fast: 0 0 0 0 1 1 1
Front edge off slow: 0 0 0 0 0 1 1

                    3     6     9
Back edge normal:   1 1 1 1 1 1 0
Back edge on slow:  1 1 1 1 1 0 0
Back edge on fast:  1 1 1 1 0 0 0
Back edge off fast: 1 1 1 0 0 0 0
Back edge off slow: 1 1 0 0 0 0 0
```



Confidence	Front Edge	Back Edge
assumption	Normal	Normal
assumption	Off/fast ramp	Normal
assumption	Normal	Normal
assumption	On/slow ramp	Normal
assumption	Normal	Normal
assumption	Normal	Normal
assumption	Normal	Normal
assumption	Normal	Normal

Setup

Before time expression data can be entered, the table layout must be specified using the setup dialog, which is accessed off the Main menu by choosing **Data->Time Course Data->Setup Time Course Table**:



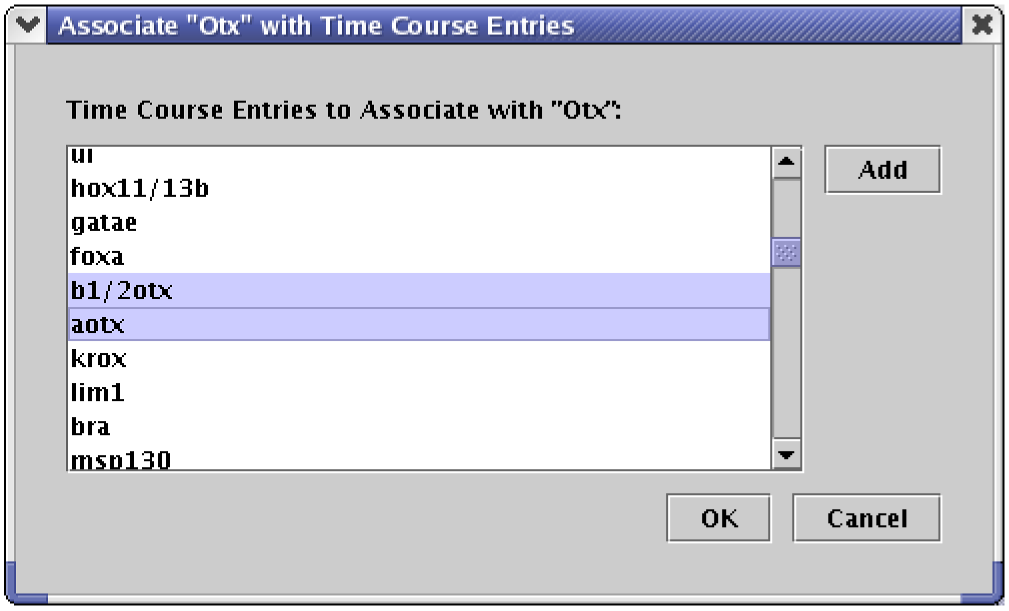
Time (hours)	Region
0	MAT
6	P
6	EM
6	EC
9	P
9	EM
9	EC
12	P

The same table is used for every gene. For each entry, you need to specify an integer time in hours, and a region name. The entries must be nondecreasing (i.e. each entry time is greater than or equal to the entry before it.) Also, once a region appears at a given time, it must appear at all subsequent times until the last time it appears.

When you add an entry, it appears before the currently selected entry row. Entries are appended to the end if a row is not selected. (If an entry is currently selected and you want to add an entry to the end, press the CTRL key while clicking on the selected entry to deselect it.)

Data Mapping

Data mapping for time expression data works much like QPCR mapping. If you require advanced control over the data associations, you can select **Manage Time Course Data->Custom Association to Data...** from the right click menu to get the following dialog box:

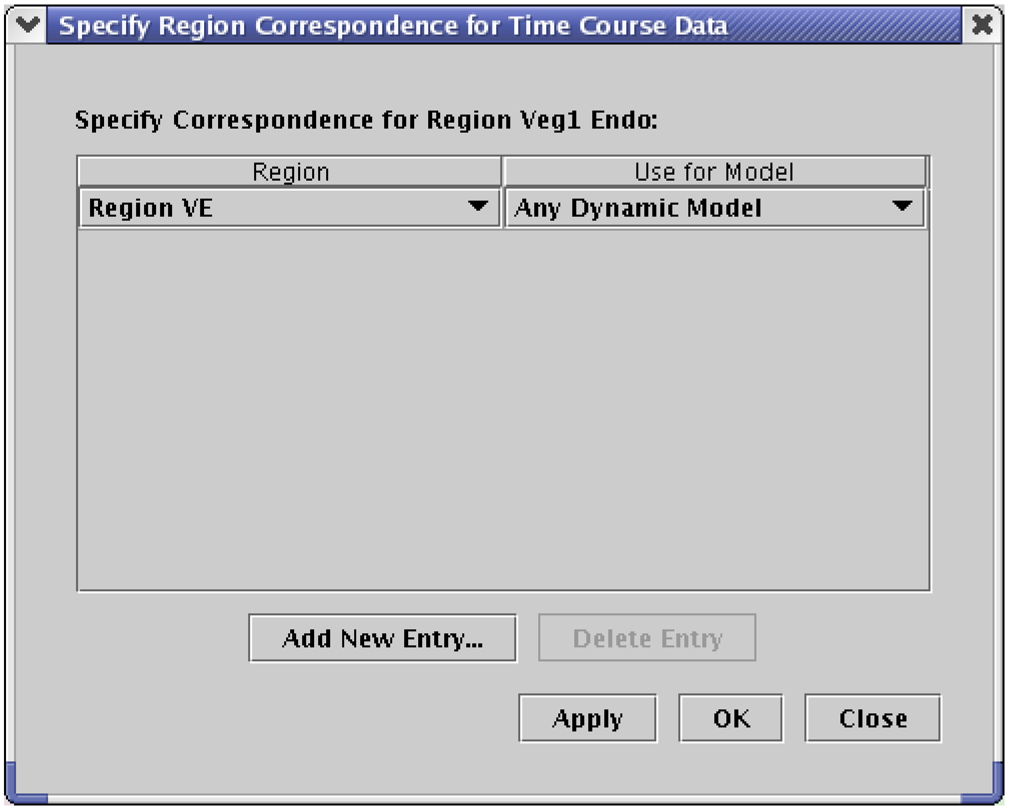

Associate "Otx" with Time Course Entries
Time Course Entries to Associate with "Otx":
ui
hox11/13b
gatae
foxa
b1/2otx
aotx
krox
lim1
bra
msp130
Add
OK
Cancel

Unlike QPCR data, where gene references can appear in multiple tables as perturbation rows, this dialog has only one list to select from. However, the selection mechanism is the same as for QPCR: the selections in the list indicate which time expression tables to associate with the gene in the network, while the **Add** button is just used to create a new entry in the list. To select or unselect multiple entries in a list, hold down the **Shift** key or the **Ctrl** key while clicking on the entry.

Region Mapping

Unlike QPCR data, time expression data also references regions; the time expression tables describe the regions and times that a gene is expressing. Thus, regions in the network model need to be associated with region tags in the time expression table. As with other data associations, a region named **PMC** shown in the network diagram is automatically matched with data tagged with **pmc**. However, for example, to match the **Veg1 Endo** region in the model with the **VE** region in the time expression data, you will need to bring up the custom association dialog by right clicking in an unoccupied area

of a region and then selecting **Manage Time Course Data->Edit Region Association...** from the menu:



Specify Correspondence for Region Veg1 Endo:

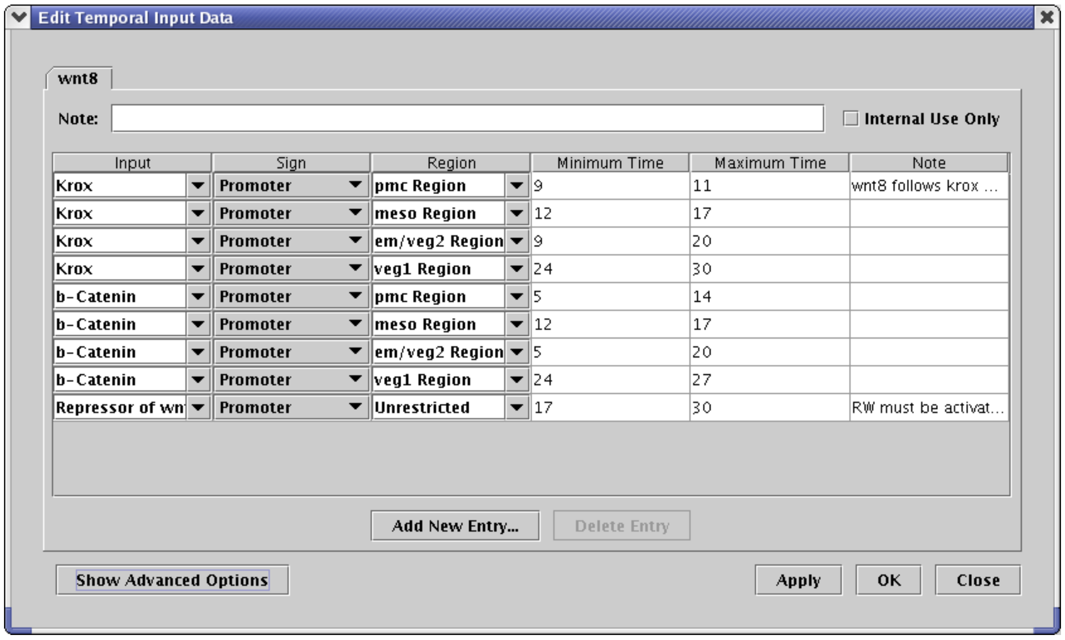
Region	Use for Model
Region VE	Any Dynamic Model

From the drop-down menu in the left column of the table, choose the region in the time expression data to associate with the region in the network. In some cases, one region in the network might be used to display more than one region described in the time expression tables, so you can add more rows to the column. In other cases, a region in the network might be used to display data from different regions, and this association might **be different for different submodels!** In that case, you select which model each row in the table refers to. Most of the time, you will leave this at **Any Dynamic Model**.

Temporal Input Data

Temporal input data describes the inputs to a gene and the times when each input is active. These data entries are used to decide when links in the network are shown as active or inactive in a dynamic model display. Currently, inputs are either on or off; there is no quantification of the level of influence of an input.

The input data can be edited using the editing dialog, which is accessed off the main menu by selecting **Data->Temporal Input Data->Manage Full Temporal Input Data Table** and then choosing an entry to edit, or by right clicking on the gene and selecting **Manage Temporal Input Data->Edit Data...**:



Note:

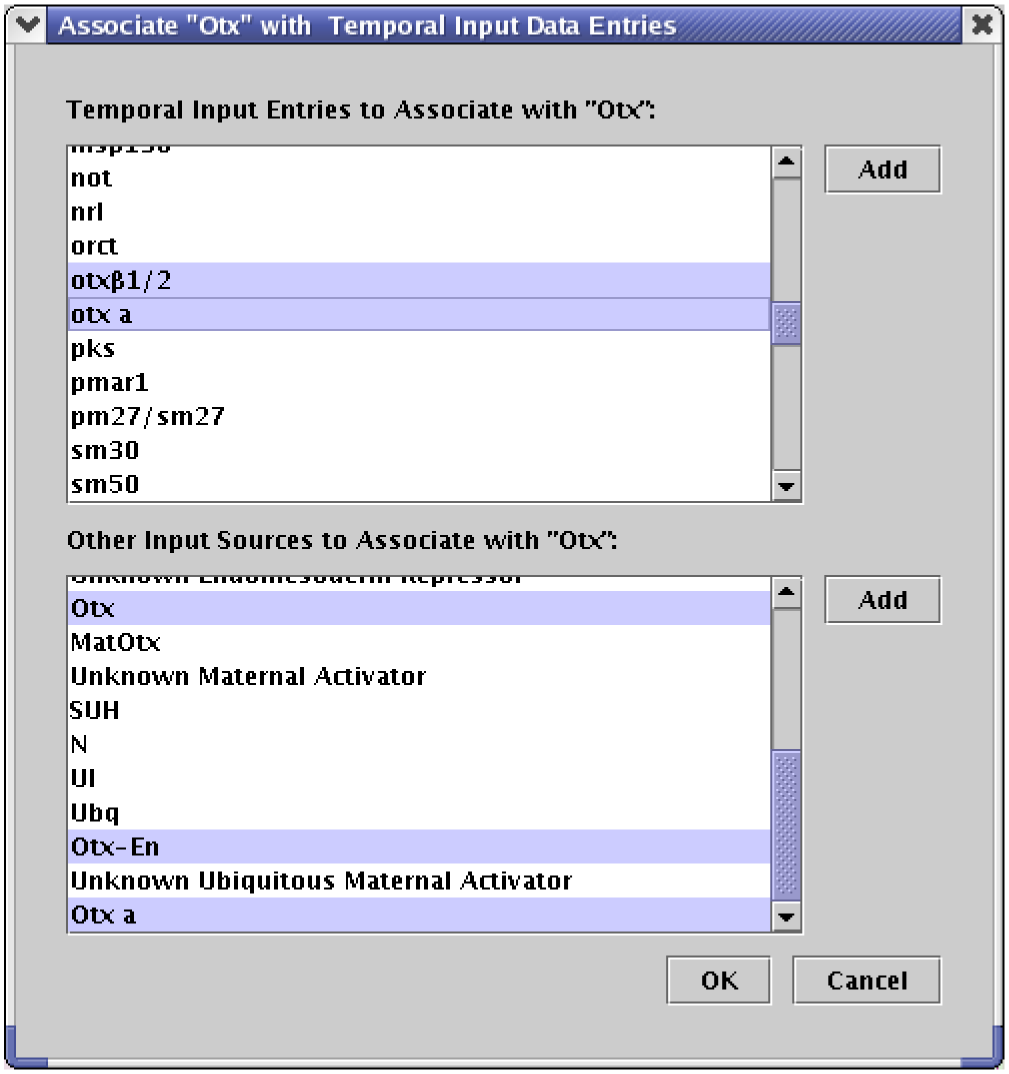
Input	Sign	Region	Minimum Time	Maximum Time	Note
Krox	Promoter	pmc Region	9	11	wnt8 follows krox ...
Krox	Promoter	meso Region	12	17	
Krox	Promoter	em/veg2 Region	9	20	
Krox	Promoter	veg1 Region	24	30	
b-Catenin	Promoter	pmc Region	5	14	
b-Catenin	Promoter	meso Region	12	17	
b-Catenin	Promoter	em/veg2 Region	5	20	
b-Catenin	Promoter	veg1 Region	24	27	
Repressor of wn	Promoter	Unrestricted	17	30	RW must be activat...

In the table, you can add a new input and set the minimum and maximum times for that input. Frequently, a temporal input entry is tagged with a particular region, since the start time and length of an input to a gene are often different in different regions. This tagging is done using the drop-downs in the third column. However, since a region does not have to be restricted if the meaning is unambiguous, the entry can be left as **Unrestricted**.

If the user selects the **Show Advanced Options** button, he can then specify the edge interpolation strategies, which modifies how the input turns on or off at the leading and trailing edges.

Data Mapping

Custom data mapping for temporal input data works much like QPCR and time expression custom mapping. You can create and edit data associations by selecting **Manage Temporal Input Data->Edit Association to Data...** from the right click menu, which launches the following dialog box:


Associate "Otx" with Temporal Input Data Entries

Temporal Input Entries to Associate with "Otx":

not
nrl
orct
otxβ1/2
otx a
pks
pmar1
pm27/sm27
sm30
sm50

Other Input Sources to Associate with "Otx":

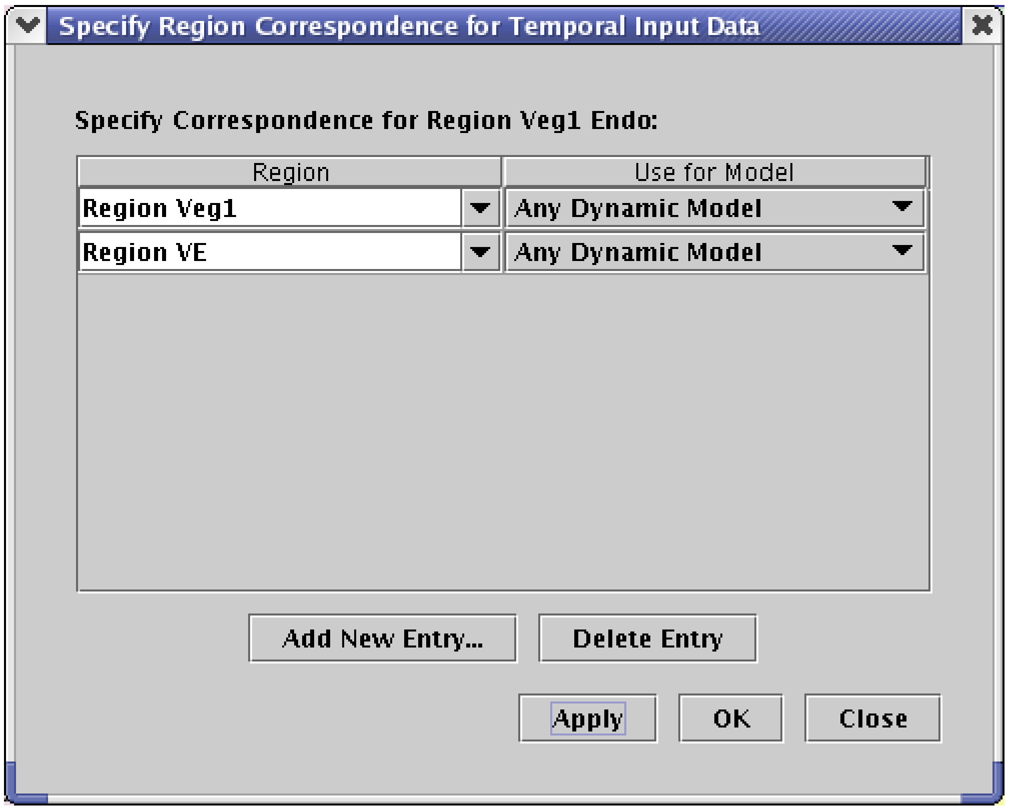
Otx
MatOtx
Unknown Maternal Activator
SUH
N
Ul
Ubq
Otx-En
Unknown Ubiquitous Maternal Activator
Otx a

Temporal input data is like QPCR data, in that this dialog has two separate lists. The first list sets an association between the network gene and one or more temporal input tables. The second list is needed because the gene can also appear in a row of other tables as an input source. In fact, nodes with no inputs will have an association specified in **only** the second list.

This dialog works like the others described above: the selections in the list indicate what temporal input elements to associate with the gene in the network, while the **Add** button is just used to create a new entry in the list. To select or unselect multiple entries in a list, hold down the **Shift** key or the **Ctrl** key while clicking on the entry.

Region Mapping

Temporal input data is like time expression data: regions in the network need to be associated with the regions defined in the data tables, since temporal inputs can vary between regions. Again, if regions are named the same in both the network and the data tables, BioTapestry automatically links the two. But to set up a customized region association, right click in an unoccupied area **of a region** and then select **Manage Temporal Input Course Data->Edit Region Association...** from the menu:



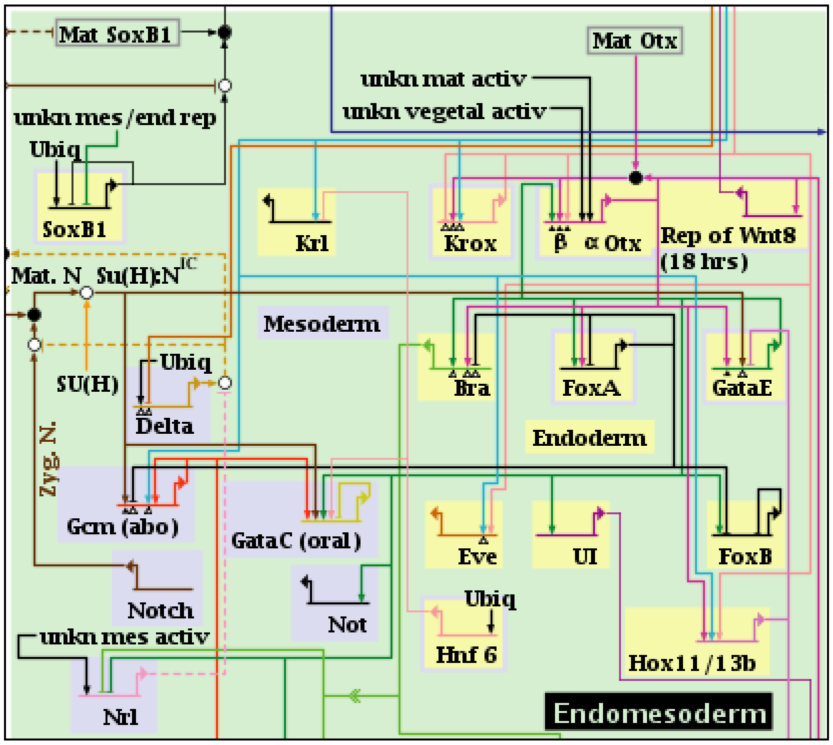
Specify Correspondence for Region Veg1 Endo:

Region	Use for Model
Region Veg1	Any Dynamic Model
Region VE	Any Dynamic Model

From the drop-down menu in the left column of the table, choose the region in the temporal input data to associate with the region in the network. In some cases, one region in the network might be used to display more than one region listed in the temporal input tables, so you can add more rows to the column. In other cases, a region in the network might be used to display data from different regions, and this association might **be different for different submodels!** In that case, you select which model each row in the table refers to. Most of the time, you will leave this at **Any Dynamic Model**.

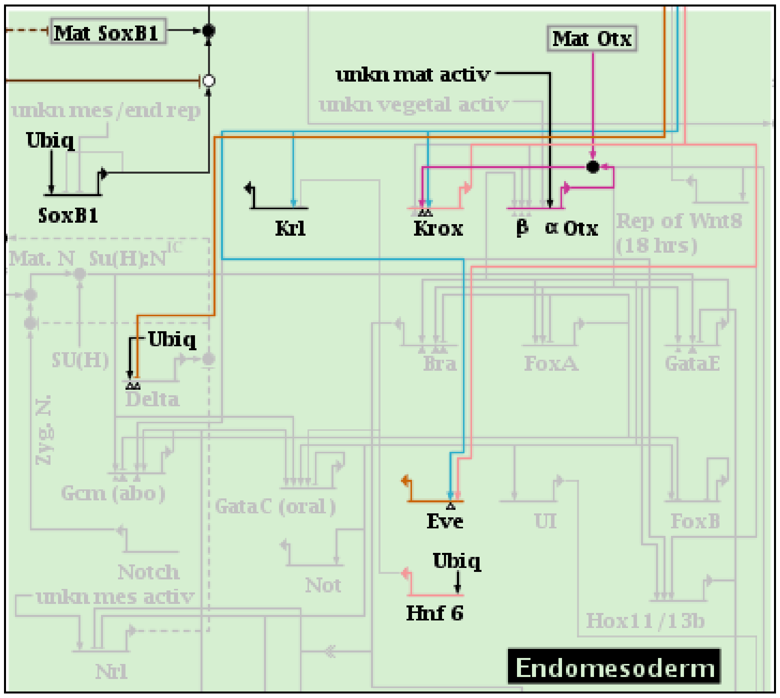
Setting Up Subregions

Every node or gene in a top-level instance model belongs to one and only one regular region. However, BioTapestry also allows the user to define any number of **subregions** within a regular region, where a node or gene in the region can belong to any number of subregions. In the following piece of the endomesoderm model, the Endomesoderm region has two defined subregions: the Endoderm and the Mesoderm.

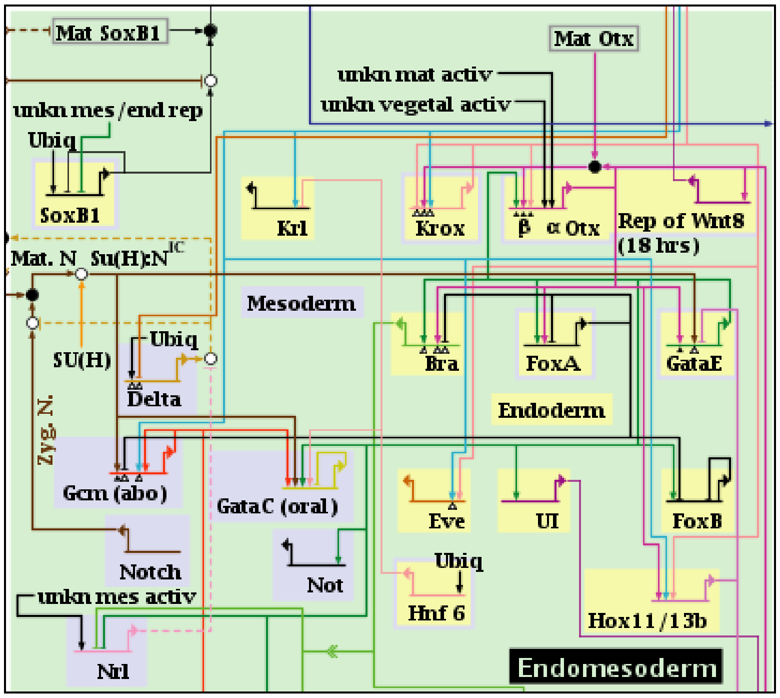


Subregions are created **and managed** by right clicking on the regular parent region and selecting the **Manage Subregions** menu. They can only be created in the top-level instance model; to create a subregion, choose **New Subregion...** from that menu.

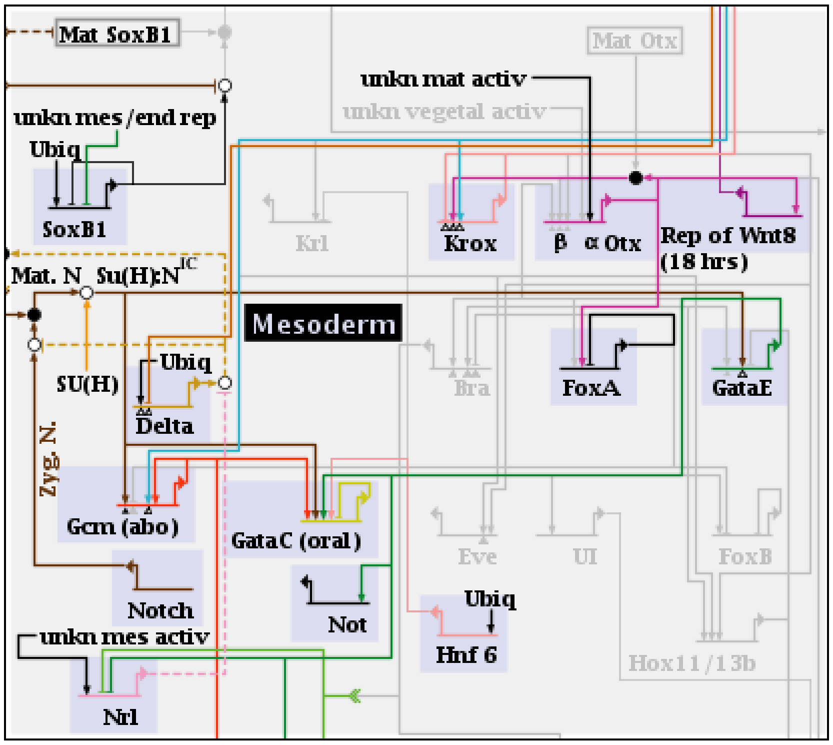
In subset models, the **New Subregion...** menu option is replaced by the two options **Activate Subregion** and **Include Subregion**. The difference between the two choices can be understood from the following pictures. The first picture shows what a region with two defined subregions looks like when it is first propagated down into the submodel. No subregions are shown:



After choosing to **include** the two subregions, they are both shown as well as the parent region:



If instead, one of the subregions is **activated**, the parent region is grayed out. Only one subregion can be activated, and it must be deactivated before an alternate subregion can be activated instead:



Regardless of whether a subregion is in the top-level instance, or is included or activated in a subset model, it is managed by right clicking on the parent region and selecting the **Manage Subregions** menu. The choices provided for managing subregions are very similar to those used for regular regions.

Editing Properties

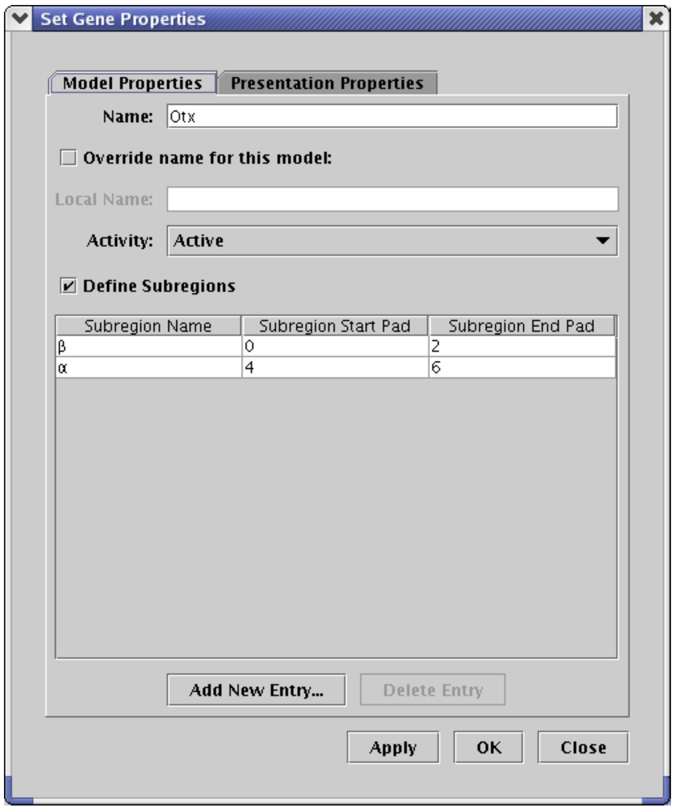
Once submodels, genes, nodes, links, and regions have been created, you can modify their properties using properties dialogs. These dialogs can be accessed by selecting the **Properties...** choice in the right-click menu.

The dialogs for genes, nodes, and links each have two tabs: one for **Model Properties** and one for **Presentation Properties**. Model properties refer to characteristics of the underlying network (e.g. is a link a promoter or a repressor), while presentation properties refer to the appearance of an element (e.g. does a gene face left or right). When editing underlying model properties, these changes will take effect across every model, since model properties are shared across all the models. For example, if you edit the region table for a gene from a top-level instance model, this change propagates up to the root model and to sibling instance models.

Presentation properties have a different effect, since the top-level instance models all have different model layouts from each other, and the root model has yet another model layout. So changes to presentation properties only affect either a top level instance model and all its submodels, or the root model, depending on the context.

Editing Gene and Node Properties

The gene and node properties dialogs look similar, except that the gene dialog has an additional subregion table for defining regions of the gene. To define gene regions, you check the **Define Subregions** box and then add a row to the table for each region:



Name: Otx

☐ Override name for this model:

Local Name:

Activity: Active

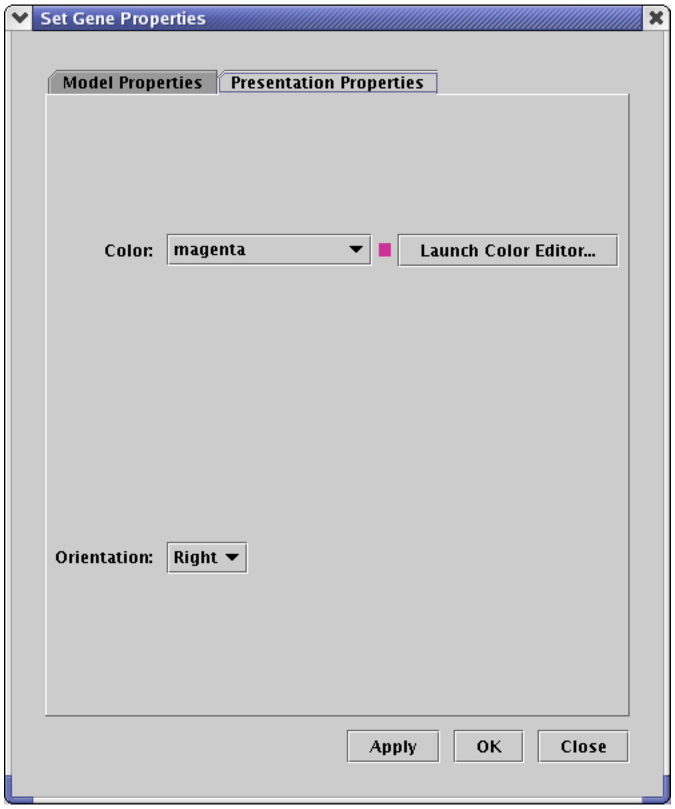
☑ Define Subregions

Subregion Name	Subregion Start Pad	Subregion End Pad
β	0	2
α	4	6

The dialog shown is for a top-level instance model. Compared to the dialog for the root model, there is an additional option to change the name of the gene for the local model, to allow for a variation of the root name that may be more appropriate for a particular time or region. This is the one case where changes on the model tab do not propagate to all other models.

The top-level instance property dialog also allows you to set the **Activity Level** for the gene or node. There are three settings: **Active**, **Inactive**, and **Vestigial**. If the element is labeled as inactive, both the name and object are rendered light gray; if labeled vestigial, the name is rendered in black, but the object is light gray.

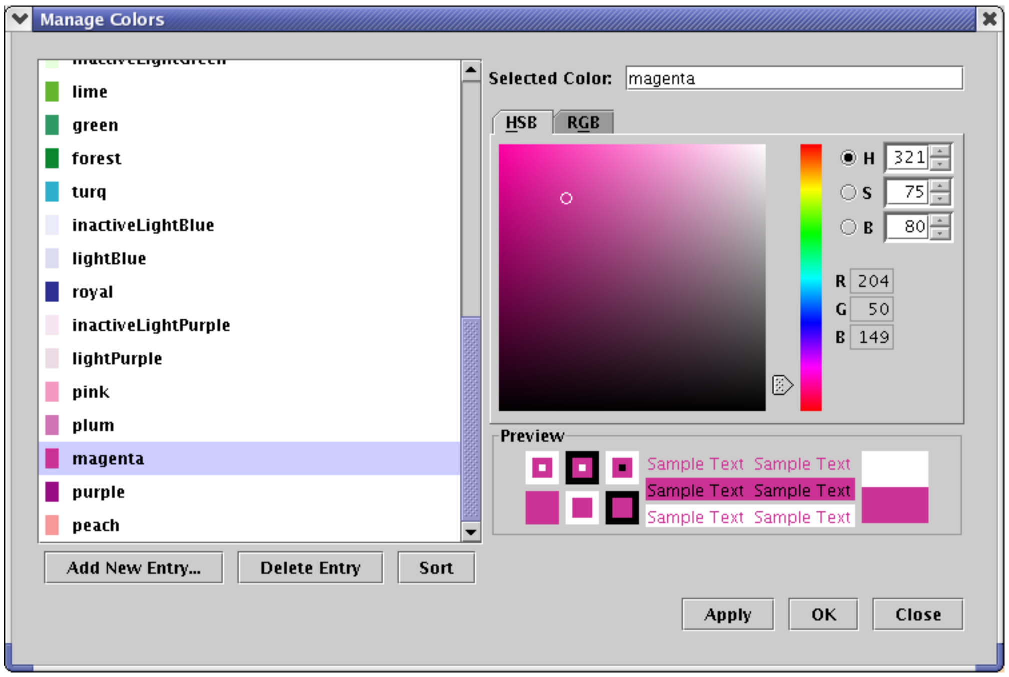
The other tab allows you to change the presentation properties:



Color: magenta

Orientation: Right

In the case of genes, this tab allows you to set the color of the gene, as well as the orientation (i.e. does the gene point left or right). When setting the color, you can select from a predefined list of colors to use; these colors are shared between all the visual elements of the network. If you wish to create a new color, or change the definition of a color (where such a change would apply to all other elements using that color), you can click on the "Launch Color Editor..." button. This brings up the color editor, which can also be selected using **Edit->Edit Colors...** from the main menu:



lime
green
forest
turq
inactiveLightBlue
lightBlue
royal
inactiveLightPurple
lightPurple
pink
plum
magenta
purple
peach

Selected Color: magenta

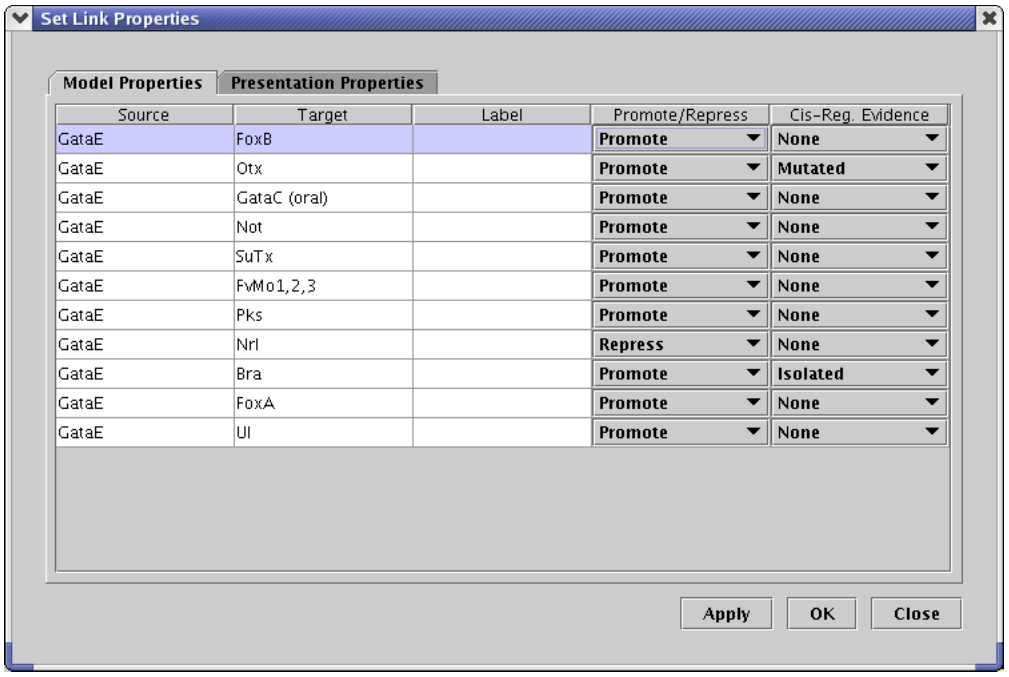
[✓ H] 321
S 75
B 80

R 204
G 50
B 149

With this dialog, you can create new colors, or edit an existing color by selecting it from the list on the left and then editing its RGB or HSV value using the tabbed editing panel on the right. The **Sort** button below the listing can be used to reorganize the list as colors are changed.

Editing Link Properties

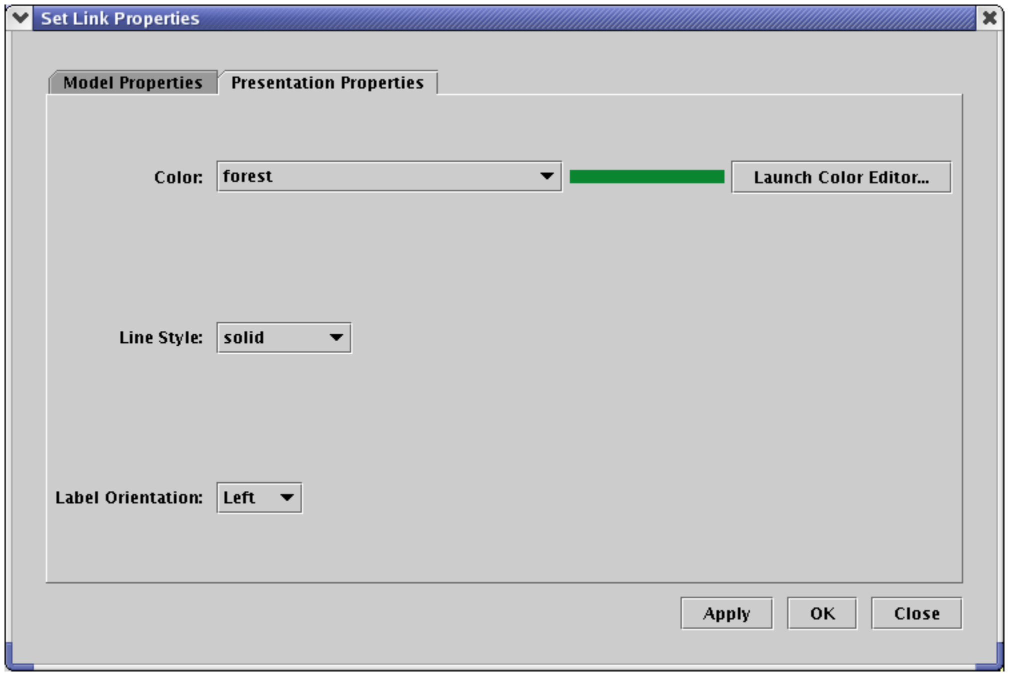
The link properties dialog also has separate tags for model and presentation properties. The difference here is that by clicking on a segment of a link tree, you are actually able to edit the properties of all the links originating from the common source node of the tree. Each separate row of the table is for a different link in the model:



Source	Target	Label	Promote/Repress	Cis-Reg. Evidence
GataE	FoxB		Promote	None
GataE	Otx		Promote	Mutated
GataE	GataC (oral)		Promote	None
GataE	Not		Promote	None
GataE	SuTx		Promote	None
GataE	FvMo1,2,3		Promote	None
GataE	Pks		Promote	None
GataE	Nrl		Repress	None
GataE	Bra		Promote	Isolated
GataE	FoxA		Promote	None
GataE	Ul		Promote	None

While the source and target columns cannot be edited, you can change the link label, the sign of the link (**Promote**, **Repress**, or **Neutral**), or a tag for the cis-regulatory evidence (**None**, **Isolated**, or **Mutated**). The cis-regulatory values are graphically represented by triangles shown opposite the end of a link (no triangle, a white triangle, or a black triangle, respectively). **NOTE**: In version 1.0 of BioTapestry, all link labels on the link tree are concatenated into a single tag for the entire tree, so there is currently a practical limit of one label per link tree.

The presentation properties tab is like that for nodes and genes:



Color: forest

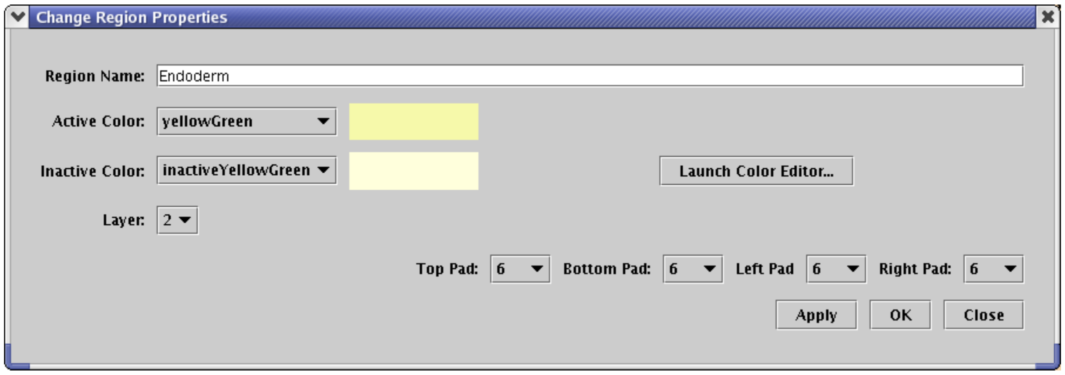
Line Style: solid

Label Orientation: Left

The difference from the gene and node properties tab is that the orientation choice refers to the link label, and you can additionally set the link line style to **solid**, **thin solid**, **dashed**, or **thin dashed**. For a link tree, you also have the option of setting special line properties for individual segments of the tree. That dialog is available by right clicking on a link segment and choosing **Change Link Segment Line Style** from the menu.

Editing Region Properties

For regular regions, right click in an unoccupied part of the region and select **Properties...**. For subregions, right click on the parent region and select **Manage Subregions->[Subregion Name]->Properties...**. This brings up the region properties dialog (this one is for a subregion):


Change Region Properties

Region Name: Endoderm

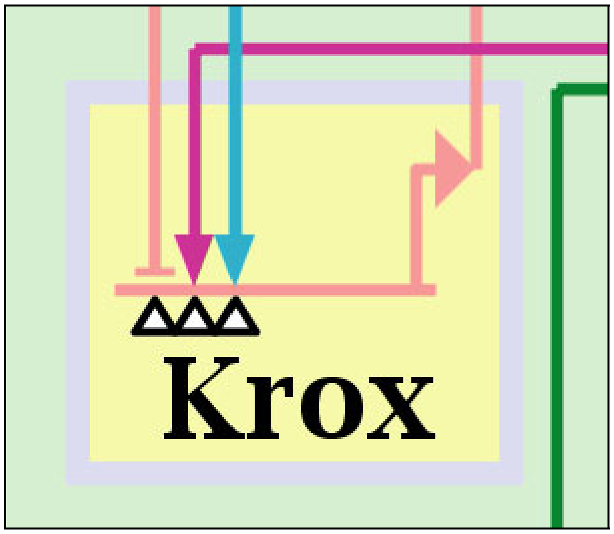
Active Color: yellowGreen

Inactive Color: inactiveYellowGreen

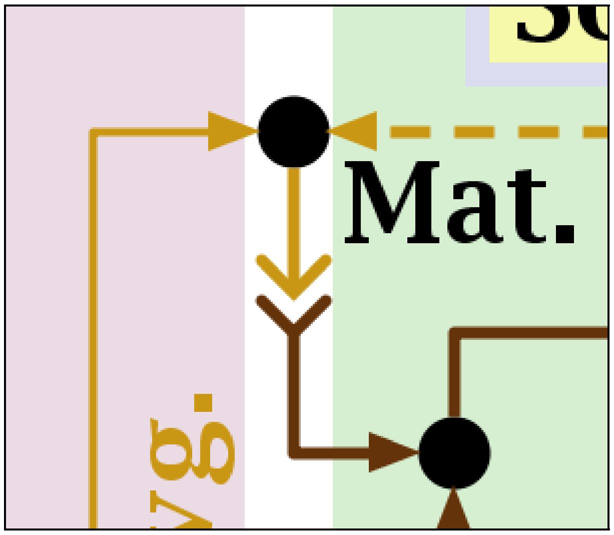
Layer: 2

Top Pad: 6 Bottom Pad: 6 Left Pad 6 Right Pad: 6

With this dialog, you can change the region name and colors; the inactive color is what is shown if you toggle a region. The **Layer** choice is shown for subregions only; it specifies which subregion is drawn below the other. This setting, in combination with the **Pad** values, allows you to show multiple subregion memberships:

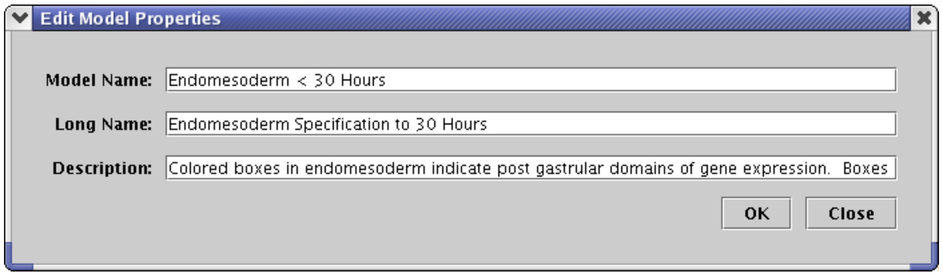


Regions are sized automatically by drawing a bounding rectangle around all the nodes and genes in the region. Since this includes intercellular nodes, you can set the various **Pad** values to negative numbers to make the edge nodes appear outside of the region boundaries:



Editing Model Properties

Right-clicking on a model name in the left-hand tree display and selecting **Edit Model Properties...** brings up a dialog for editing the model properties. The actual dialog depends on whether the model is a dynamic or static submodel. For static submodels, you get to change the name shown in the tree display, the title name shown in the network diagram, and the description that appears in the text display at the bottom of the BioTapestry window:

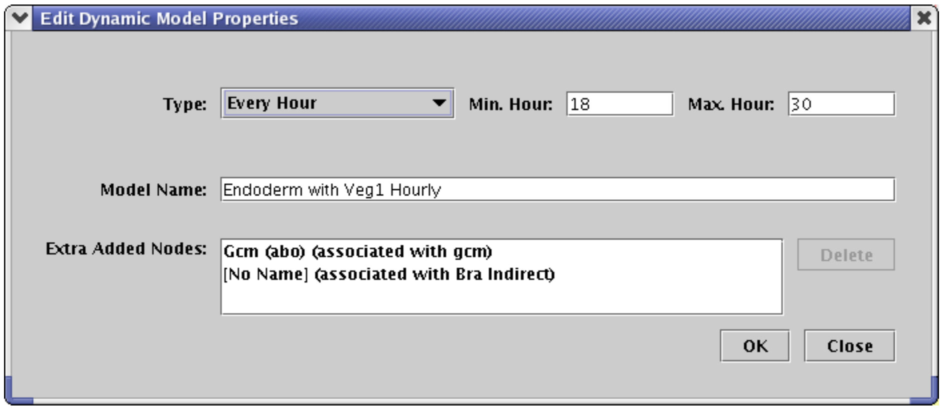

Edit Model Properties

Model Name: Endomesoderm < 30 Hours

Long Name: Endomesoderm Specification to 30 Hours

Description: Colored boxes in endomesoderm indicate post gastrular domains of gene expression. Boxes

For dynamic submodels, the user can change the model type and time bounds, although these options become more restricted once submodels have been defined:


Edit Dynamic Model Properties

Type: Every Hour   Min. Hour: 18   Max. Hour: 30

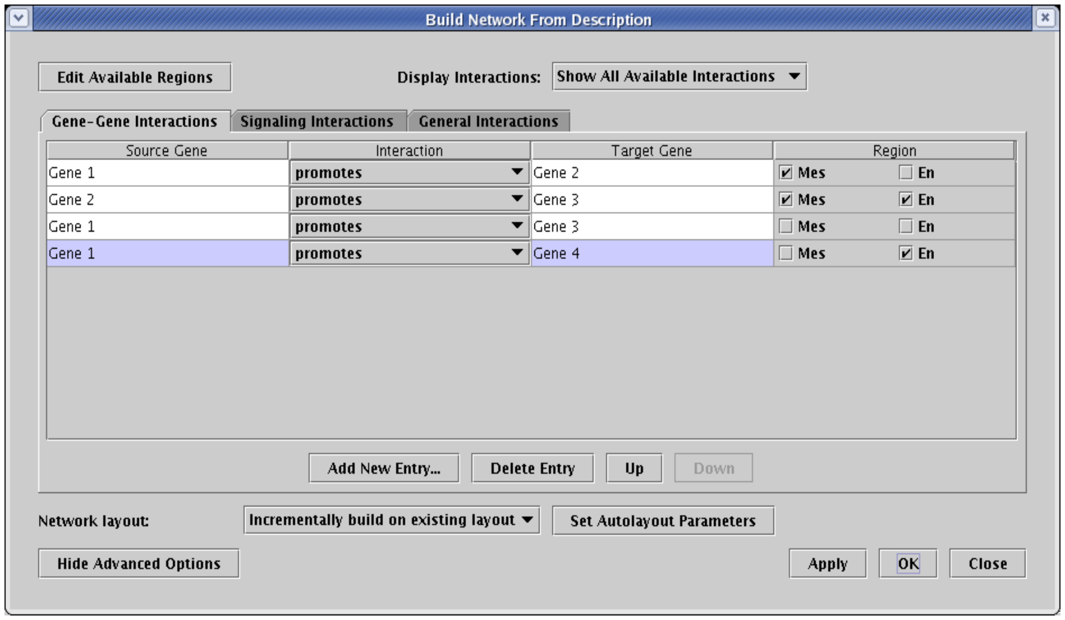
Model Name: Endoderm with Veg1 Hourly

Extra Added Nodes:
Gcm (abo) (associated with gcm)
[No Name] (associated with Bra Indirect)

This dialog is also where you can edit any extra nodes that have been added to the dynamic submodel beyond those that belong to the regions that are explicitly included in the model.

Building a Model Hierarchy Using Interaction Tables

One way to create a model hierarchies is by drawing the root network, and then propagating selected elements down into submodels. That approach was covered in the first portion on this guide. However, BioTapestry can also create a model hierarchy automatically from lists of interactions. To use this feature, select any model in a hierarchy, then select **Tools->Build Network From Description...** from the main menu. A dialog box that allows you to specify each interaction in the network appears:



Display Interactions: Show All Available Interactions

Source Gene	Interaction	Target Gene	Region	
Gene 1	promotes	Gene 2	☑ Mes	☐ En
Gene 2	promotes	Gene 3	☑ Mes	☑ En
Gene 1	promotes	Gene 3	☐ Mes	☐ En
Gene 1	promotes	Gene 4	☐ Mes	☑ En

Network layout: Incrementally build on existing layout

While this approach has some limitations compared to drawing (e.g. you cannot have an arbitrary nameless node), using interaction tables to define a network currently has some advantages over drawing:

1) The network hierarchy can be built from the "bottom up".

2) Nodes and regions will be automatically laid out for you.

3) Some interactions are represented by higher-level abstractions, i.e. as network motifs. This currently only applies to signals, but this principal can be extended in the future.

Top Down Versus Bottom Up

The technique of creating model hierarchies by drawing the root network and then propagating selected elements down into submodels can be thought of as *top-down*. An alternative way of building a hierarchy is *bottom-up*. The networks at the bottom of the model hierarchy are created first, and BioTapestry automatically creates the models at the higher levels by building the union of all the features in these subset models. Building from interaction tables supports both approaches. If you first create tables for the root model, then you can specify which entries should be included in a submodel, working from the top down. You can also go directly to the models at the bottom of the hierarchy and populate those tables first; the tables for models higher up will then be filled in automatically.

The Table Entries are Definitive

When tables of interactions are used to create a network, that description of the network topology is the definitive one. Thus, adding new genes, nodes, links or regions by *drawing* them or propagating them does not add them to the tables, nor does deleting them from the drawing remove them. Any such changes will disappear the next time the network is rebuilt from the table definitions. If you happen to make changes to the network *layout* (e.g. moving nodes or rerouting links), these can be retained if you desire, but you need to add or remove interactions *using the tables* to make any lasting changes to the network definition.

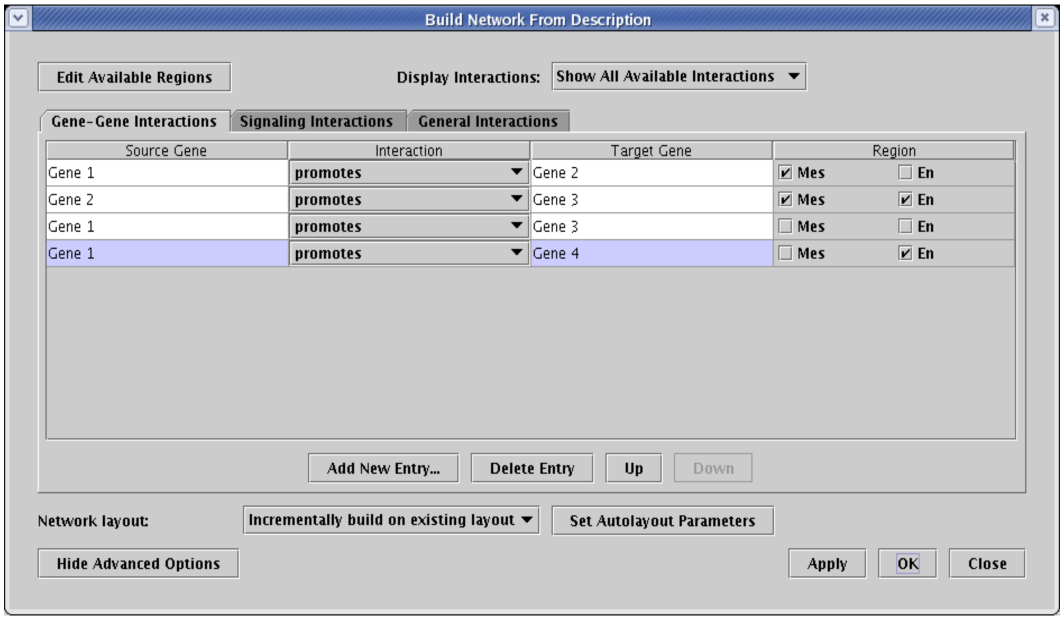
Thus, once you choose to build a network from interaction tables, you should continue to use this method to build and modify the network definition. If you wish to abandon this approach and just use drawing thereafter, you can select **Tools->Drop All Interaction Tables Used to Build Networks** from the main menu to make the drawn versions of the networks be the definitive definitions.

(**NOTE**: It is anticipated that a future version of BioTapestry will relax this restriction and support a bidirectional information flow between drawing operations and the interaction entries).

Each Tab Represents Different Network Motifs

Looking at the network building dialog shown above, you see that there are three different tabs: one for **Gene-Gene Interactions**, one for **Signaling Interactions**, and one for **General Interactions** tab. (This last tab only appears if you have clicked on the **Show Advanced Options** button.) Each of these three tabs represents a different type of interaction.

1) The **Gene-Gene Interactions** tab handles the most common case, where one gene promotes or represses another gene in the same region:

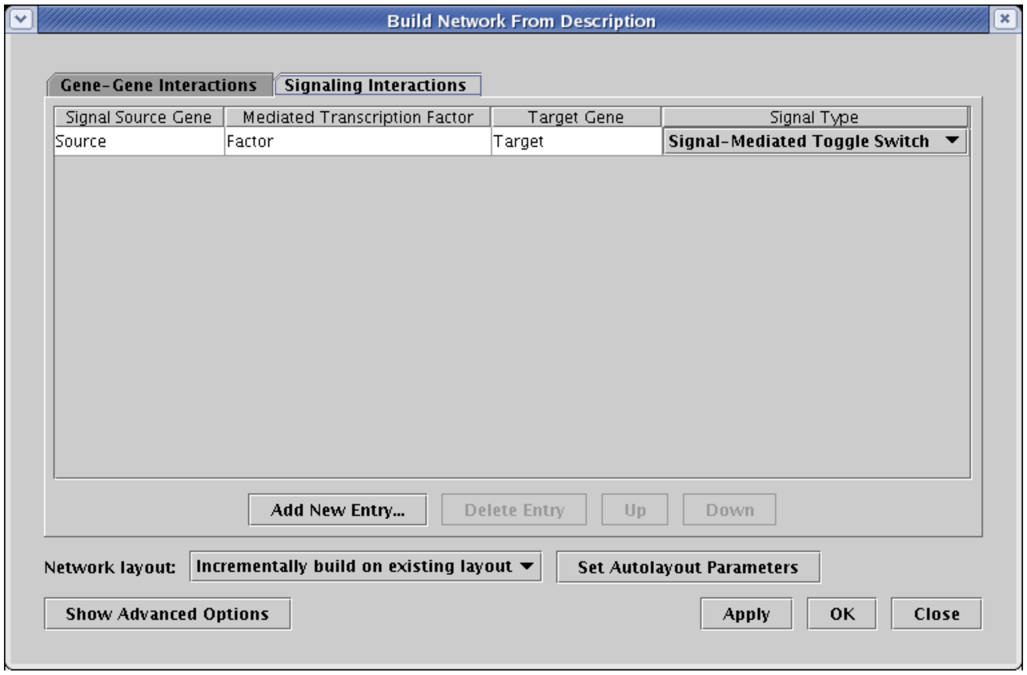


Build Network From Description
Edit Available Regions
Display Interactions: Show All Available Interactions
Gene-Gene Interactions
Signaling Interactions
General Interactions

Source Gene	Interaction	Target Gene	Region	
Gene 1	promotes	Gene 2	✓ Mes	☐ En
Gene 2	promotes	Gene 3	✓ Mes	✓ En
Gene 1	promotes	Gene 3	☐ Mes	☐ En
Gene 1	promotes	Gene 4	☐ Mes	✓ En

Add New Entry...
Delete Entry
Up
Down
Network layout: Incrementally build on existing layout
Set Autolayout Parameters
Hide Advanced Options
Apply
OK
Close

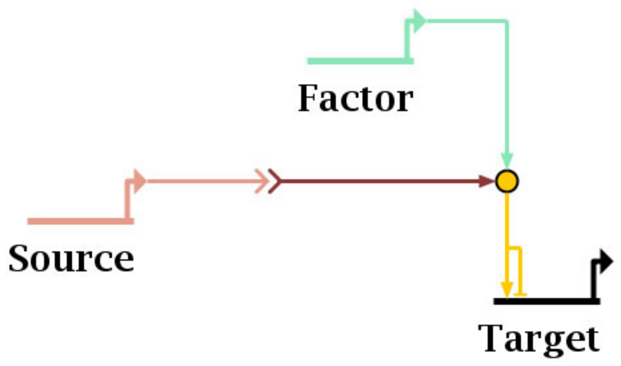
2) An entry in the table on the **Signaling Interactions** tab represents a signaling interaction between two different regions, and corresponds to a network motif consisting of five different nodes. For example, the following root-level table entry:



Signal Source Gene	Mediated Transcription Factor	Target Gene	Signal Type
Source	Factor	Target	Signal-Mediated Toggle Switch

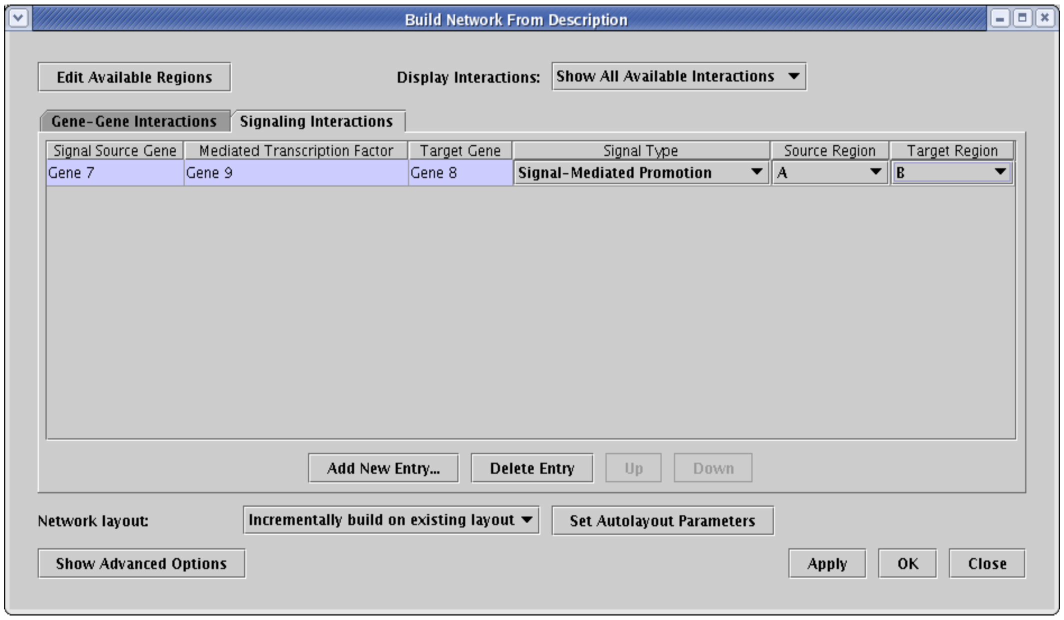
Network layout: Incrementally build on existing layout

creates the network motif shown below. The **Signal-Mediated Toggle Switch** setting used here for **Signal Type** creates two links inbound to the target, indicating the effect of the transcription factor as a promoter or repressor depends on the signal state:



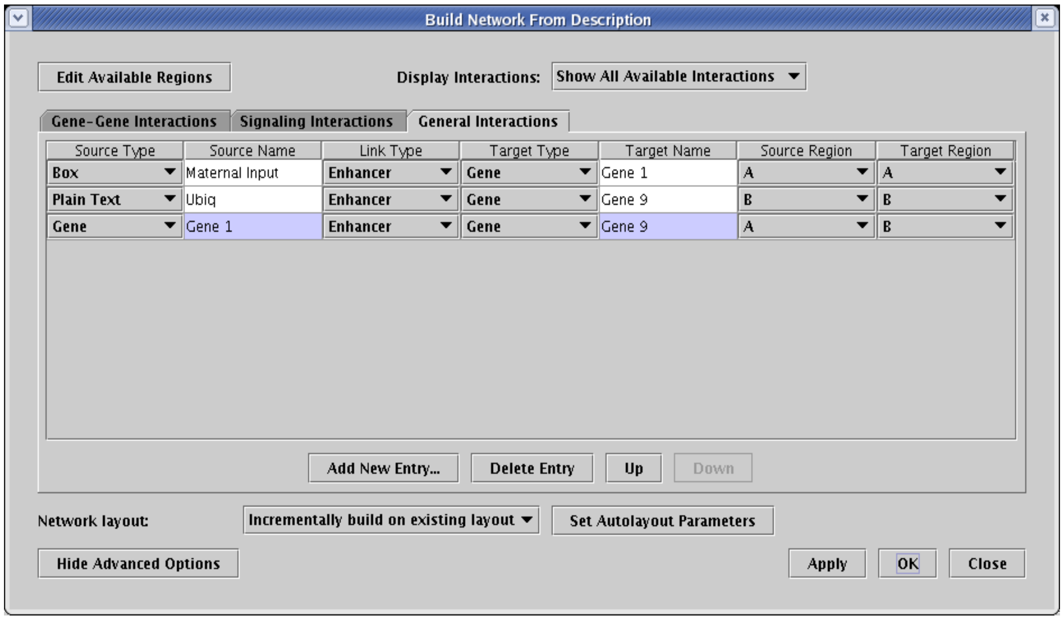
The other two possible **Signal Type** settings create either a single promotion or repression link.

If you are not working at the level of the root model, the signal table will also contain entries to specify the source and destination regions for the signal:


Build Network From Description
Edit Available Regions
Display Interactions: Show All Available Interactions
Gene-Gene Interactions
Signaling Interactions
Add New Entry...
Delete Entry
Up
Down
Network layout: Incrementally build on existing layout
Set Autolayout Parameters
Show Advanced Options
Apply
OK
Close

Signal Source Gene	Mediated Transcription Factor	Target Gene	Signal Type	Source Region	Target Region
Gene 7	Gene 9	Gene 8	Signal-Mediated Promotion	A	B

3) The **General Interactions** tab, shown below, is available as an advanced option, and handles the general case of any type of source node interacting with any type of target node, e.g. bubbles, boxes, diamonds, etc. Also, this table allows you to specify non-signaling interactions that span across regions. (That cross-region flexibility may be important to you, since the simple gene-gene table only allows you to specify the common case of interactions that have the source and target in the same region.)



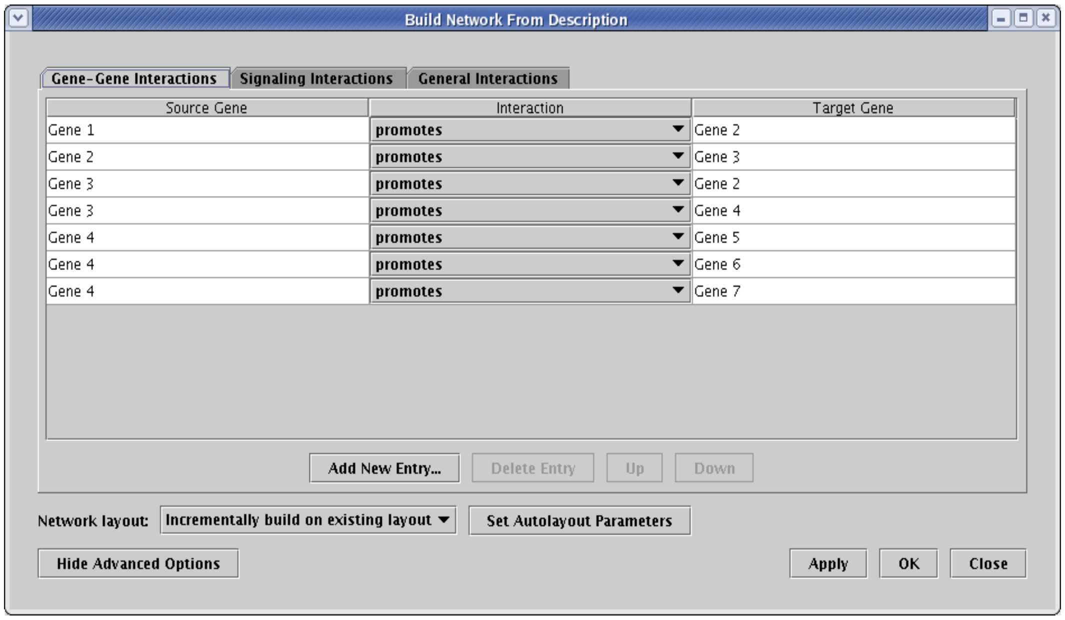
Display Interactions: Show All Available Interactions

Source Type	Source Name	Link Type	Target Type	Target Name	Source Region	Target Region
Box	Maternal Input	Enhancer	Gene	Gene 1	A	A
Plain Text	Ubiq	Enhancer	Gene	Gene 9	B	B
Gene	Gene 1	Enhancer	Gene	Gene 9	A	B

Network layout: Incrementally build on existing layout

Specifying Interactions at the Root Level

When you are working with the root level model, the interaction tables are different than for other models, since they do *not* include columns for specifying the regions for an interaction; regions are not relevant at this level of the hierarchy. At the root level, you can create a new gene-gene interaction simply by clicking the **Add New Entry...** button, then specifying the source and target genes and the type of interaction (promotion or repression):



Source Gene	Interaction	Target Gene
Gene 1	promotes	Gene 2
Gene 2	promotes	Gene 3
Gene 3	promotes	Gene 2
Gene 3	promotes	Gene 4
Gene 4	promotes	Gene 5
Gene 4	promotes	Gene 6
Gene 4	promotes	Gene 7

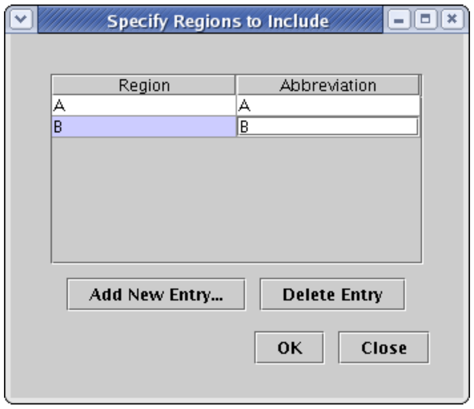
Network layout: Incrementally build on existing layout

The other two dialog tabs work in the same fashion for the root level model, i.e. without any mechanism to specify source and target regions for the interaction.

Specifying Interactions Below the Root Level

Once you start working with interaction tables below the root level, you need to start specifying regions, because all interactions at these levels are region-specific. An interaction has a source region and a target region. In the common case (gene-gene interactions), these two regions are the same. However, these regions can be different for rows created in the signaling and general interactions tables.

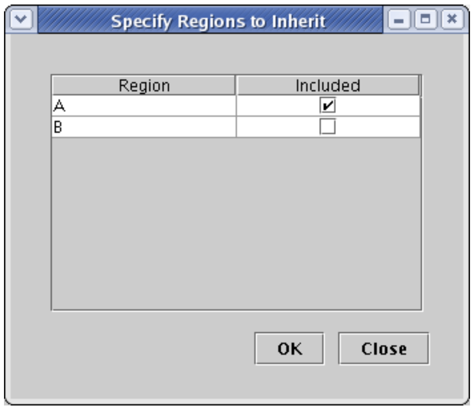
When you are working with a network model below the root level and select **Tools->Build Network From Description...**, you are prompted to specify the regions you want in that model if you have not done so before:



Region	Abbreviation
A	A
B	B

While the region name you give can be any length, you must also provide a unique three-letter (or less) abbreviation for the region. This abbreviation is used in the tables to label checkboxes.

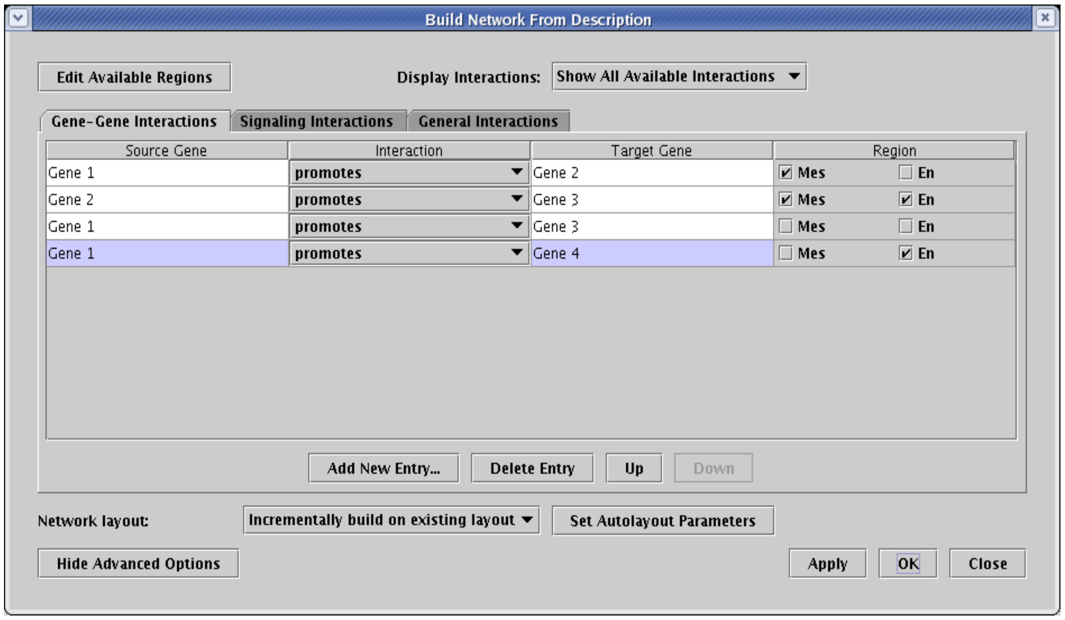
While the above region dialog appears if you are working with a model that does not have a parent model containing inheritable regions, an alternate region dialog will appear if you simply need to specify what regions to inherit from the parent model:



Region	Included
A	✓
B	☐

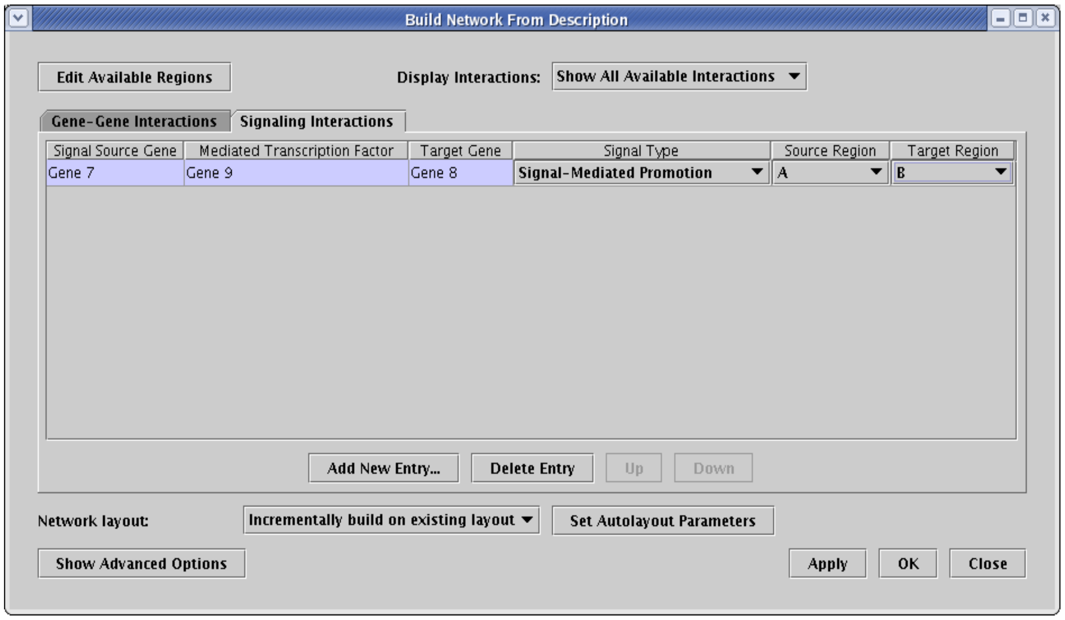
It is important to note that the regions you have available to select in the interaction tables can always be modified at any time by clicking on the **Edit Available Regions** button. Also, since the hierarchy can be built from the bottom-up, you can add a region to a child model that does *not* exist in the parent, and not just specify existing regions to inherit; new regions will be added to the parent when the child model is built.

The interaction tables for non-root models include columns for specifying regions where the given interaction is present. In the gene-gene interaction table, there is a check box for each region, and you can check all, some, or none of the regions:


Build Network From Description
Edit Available Regions
Display Interactions: Show All Available Interactions
Gene-Gene Interactions
Signaling Interactions
General Interactions
Add New Entry...
Delete Entry
Up
Down
Network layout: Incrementally build on existing layout
Set Autolayout Parameters
Hide Advanced Options
Apply
OK
Close

Source Gene	Interaction	Target Gene	Region	
Gene 1	promotes	Gene 2	✓ Mes	☐ En
Gene 2	promotes	Gene 3	✓ Mes	✓ En
Gene 1	promotes	Gene 3	☐ Mes	☐ En
Gene 1	promotes	Gene 4	☐ Mes	✓ En

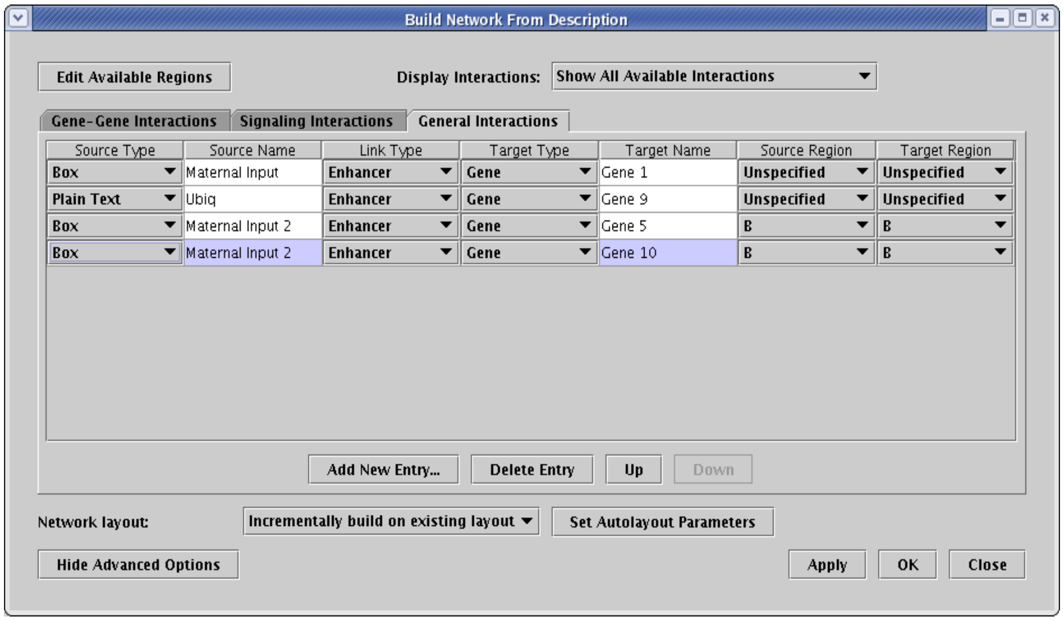
The other tabs have tables with separate columns for the source and target regions, so that they can be different:


Build Network From Description
Edit Available Regions
Display Interactions: Show All Available Interactions
Gene-Gene Interactions
Signaling Interactions
Add New Entry...
Delete Entry
Up
Down
Network layout: Incrementally build on existing layout
Set Autolayout Parameters
Show Advanced Options
Apply
OK
Close

Signal Source Gene	Mediated Transcription Factor	Target Gene	Signal Type	Source Region	Target Region
Gene 7	Gene 9	Gene 8	Signal-Mediated Promotion	A	B

The Meaning of Deletions and Rows With No Selected Regions

When working with interaction tables below the level of the root model, it is essential to understand the significance of rows in the table that have *no regions* selected, e.g. the first two rows of the following interaction table:



Display Interactions: Show All Available Interactions

Source Type	Source Name	Link Type	Target Type	Target Name	Source Region	Target Region
Box	Maternal Input	Enhancer	Gene	Gene 1	Unspecified	Unspecified
Plain Text	Ubiq	Enhancer	Gene	Gene 9	Unspecified	Unspecified
Box	Maternal Input 2	Enhancer	Gene	Gene 5	B	B
Box	Maternal Input 2	Enhancer	Gene	Gene 10	B	B

Network layout: Incrementally build on existing layout

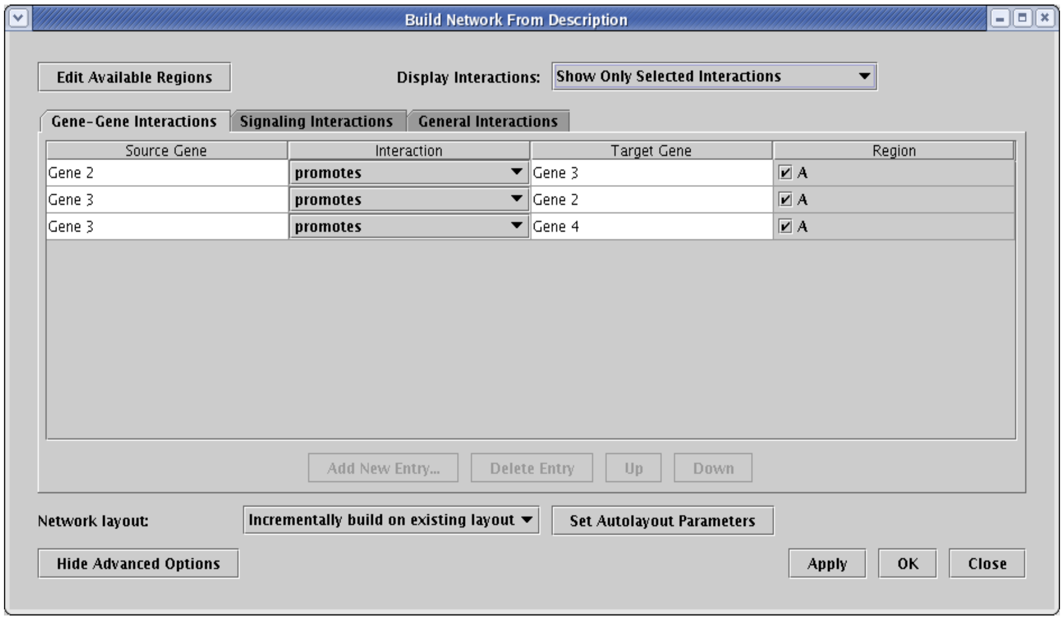
Such rows are important, since they represent interactions that still exist in the root model and possibly other models in the hierarchy. Thus, *you should not delete a row with no selected regions unless you wish to completely remove it from all models*. In a similar vein, you can add interactions that have no regions specified to a non-root model; they will only be added to the root model. In summary, you should consider the full set of interactions listed in the tables as defining the root model, regardless of the current region selections. Those rows that have selected regions specify what will appear in the current model.

Furthermore, it is important to understand the meaning of *unselecting* a region assignment when editing an existing subset model in the hierarchy. If you are thinking in a "top-down" fashion, you probably want the program to just delete this interaction from the current model and its child models. But if you are thinking "bottom-up", you want the program to also delete it from all parent models too (except, of course, for the root model; you would just delete the whole table entry if you wanted that). BioTapestry handles this deletion ambiguity by asking you to specify what you want when you remove an existing interaction from a model.

In summary, row deletions in particular can have far-reaching effects on all other models in the hierarchy, even distant cousins that are not direct ancestors, so treat them with care!

Selecting Which Interactions to View

Since all the rows of an interaction table need to be present (as discussed above), even if they are not relevant for a particular model, it becomes important to be able to filter out those interactions that you do not wish to see. The **Display Interactions** dropdown menu provides a few different options for filtering down the rows that are displayed. For example, you can just show those rows that have been selected for the current model:



Display Interactions: Show Only Selected Interactions

Source Gene	Interaction	Target Gene	Region
Gene 2	promotes	Gene 3	☑ A
Gene 3	promotes	Gene 2	☑ A
Gene 3	promotes	Gene 4	☑ A

Network layout: Incrementally build on existing layout

The following options are available:

Show All Available Interactions: This option does not filter out any interactions.

Show Only Selected Interactions: This option filters out interactions that have no regions selected in the current model.

Show Newly Added Interactions: This option shows only newly added table rows that have not yet been added to the network.

Show Interactions Present in Parent Model: This option shows only those table rows that are used in the parent model of the current model. This option is only offered if there is a parent model with interactions already present.

Network Layout Options

There are two network layout options available when building a network from a list of interactions: you can choose to either **Discard previous layout**, or **Incrementally build on existing layout**.

When you build a network model from a list of interactions, any additions or deletions are first applied to the root network layout, and then those changes are propagated down to all the models that need to incorporate those changes. So it is important to realize that choosing **Discard previous layout** can have far-reaching effects in a large model hierarchy, since the layouts of *all* other models built from interactions will be discarded and rebuilt! In contrast, if you choose **Incrementally build on existing layout**, most of the current layout will be retained intact, with network additions being arranged around the current elements of the network.

There are a few caveats to keep in mind about the auto layout:

Even if you choose to discard the layout, the system will still make an effort to retain the existing node and link colors in the new layout. Some colors may be change if the coloring algorithm is trying to avoid a link crossing ambiguity.

Even if you choose to build on the existing layout, all links *between regions* will still be laid out again. (This is due to a limitation in the current implementation of the layout engine).

If you are adding many elements to an existing network layout, and/or if the network layout is tightly compressed, it is a good idea to use the **Expand Network** tool first to add space for placing the new elements (especially links).

Layout parameters can be changed by clicking the **Set Autolayout Parameters** button. The options on that dialog are described in the section **Autolayout Tools**; that section also gives tips on using the automatic layout features.

Automatic Layout Tools

Overview

The automatic layout engine is used in a couple of different ways within BioTapestry. First, it is used to completely lay out network models, both nodes and links, from interaction tables (i.e. by invoking **Tools->Build Network From Description**). Note that since importing from SIF files or CSV files builds these interaction tables before then using the tables to build the network, this way is relevant to those cases too.

The other way that the layout engine is used is to handle link layouts only; these options are available either from the **Tools->Automatic Layout Tools** menu or the right-click popup menu for links. The crucial thing about these tools is that they are ONLY focused on providing help with laying out links: they do not provide *any* help with laying out the genes. On the plus side, these layout features are available even if you are just drawing a network (and not using interaction tables).

The upshot of this distinction is that if you want automatic node layout, the only way you get that is by using interaction tables. If you are creating a network by drawing it, then you only have auto link support. After all, it is unlikely that automatic node layout will give you what you want. Instead, you can probably do a better job of grouping and positioning genes in your diagram than the layout engine can, since you can provide the essential biological insights that will make the complex network comprehensible.

Technique

When building a network from interaction lists, the engine lays out both network nodes and links in the root model first, and then uses this basic layout as a template for creating layouts in all the submodels. A network is laid out in a left-to-right hierarchical fashion, with source nodes on the left and terminal target nodes on the right. The order of the interactions in the interaction tables determines the order in which nodes are added to the network (and also in which order feedback loops are resolved). Regions are also laid out in hierarchically from left to right, based on the ordering of source and target regions.

Link-Only Automatic Layout Options

As you drag nodes around to create your desired network organization, the attached orthogonal links will be all messed up. While you can always manually create, reposition, and delete link corners to achieve a comprehensible layout, BioTapestry provides three ways to automatically fix up the link layouts:

From main menu bar: **Tools->Automatic Layout Tools->Automatic Link Layout**. This option will layout *all* links in the network.

From main menu bar: **Tools->Automatic Layout Tools->Layout Only Irregular Links**. This option will layout only those links in the network that have non-orthogonal segments or which overlap themselves or other links. Note that this method lays the entire link tree out again, not just the crooked or overlapping portion. Also, links that overlap non-target nodes are not fixed by this operation.

From link right-click menu: **Auto Layout Links Through This Segment**. This option will layout only that portion of the link tree bound for the targets downstream of the segment you right-click on. (Note that this means all downstream segments, as well as any upstream segments that are on the path to the identical set of targets.)

Link Optimization

The automatic link layout engine uses a multi-pass algorithm, where the first pass comes up with a rough candidate layout, and all subsequent passes do clean-up and optimization. However, since this optimization pass can be slow for large networks, and particularly for a large hierarchy of several networks, it is left to the user to apply it as desired. One should understand that the result of not applying at least one cleanup pass is that the link layout may contain some bizarre artifacts. There are several ways to access the optimization function:

You can apply a single optimization pass to all links in the layout from the main menu: **Tools->Automatic Layout Tools->Run Single Link Optimization Pass**. (Links that are not laid out orthogonally are ignored by the optimization engine.)

You can apply a single optimization pass to a single link tree by right-clicking on any segment in the tree and choosing **Single Pass Optimization of Link Tree**. (Note that the optimization is applied to the entire tree).

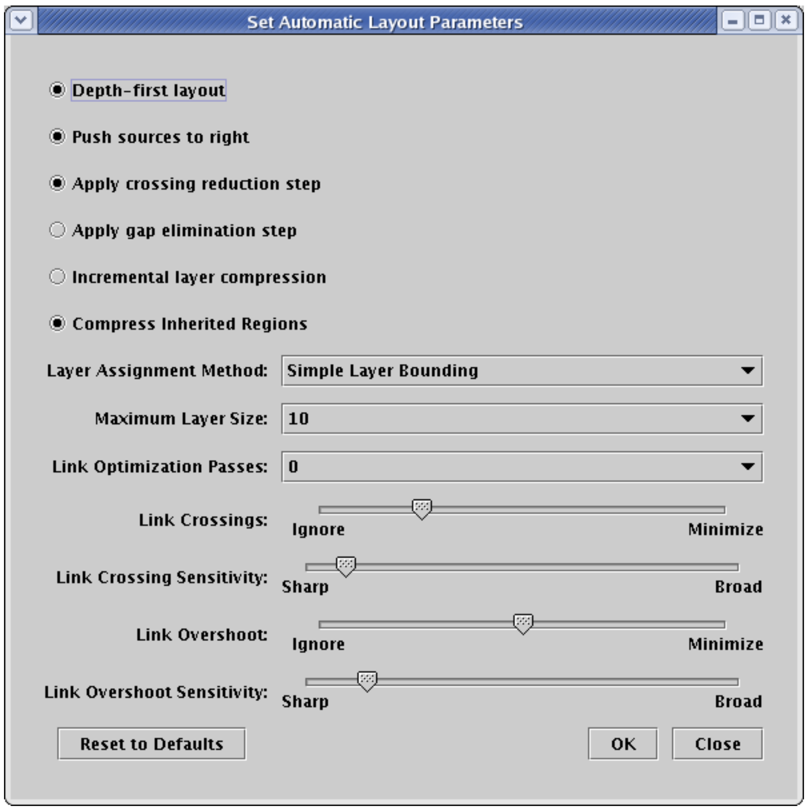
On some operations (e.g. importing CSV files), you can check a box if you desire to apply one optimization pass automatically as part of the layout process.

You can set the automatic layout options to include one or more link layout optimization passes to standard autolayout processes.

Auto Layout Parameters

The user can set several parameters that determine how the layout engine behaves. The dialog box to do this can either be accessed from **Tools->Automatic Layout Tools->Set Automatic Layout Options**, or by pressing the **Set Autolayout Parameters** button on the **Build Network From Description** dialog. Most of these parameters control how nodes and genes are laid out when networks are built from interaction tables, though a few drive link layout behavior.

The following dialog box shows the parameters that can be set. These are described below.


Set Automatic Layout Parameters

● Depth-first layout

● Push sources to right

● Apply crossing reduction step

○ Apply gap elimination step

○ Incremental layer compression

● Compress Inherited Regions

Layer Assignment Method: Simple Layer Bounding

Maximum Layer Size: 10

Link Optimization Passes: 0

Link Crossings: Ignore — Minimize

Link Crossing Sensitivity: Sharp — Broad

Link Overshoot: Ignore — Minimize

Link Overshoot Sensitivity: Sharp — Broad

Depth-first layout If this option is not checked, the layout engine adds new nodes to the network using the ordering of the interaction list. However, with depth-first layout, all interactions originating from the target of the first interaction are added recursively before the second interaction in the list is added.

Push sources to right When this option is selected, nodes which target other nodes, but which are not targets themselves, will be pushed as far right as possible in the network layout. If the option is unselected, these nodes will tend to remain on the left side of the layout.

Apply crossing reduction step If this option is selected, the nodes in each column of the left-to-right hierarchy will be swapped within the column if that reduces link crossings. The downside is that simple source-target motifs could get scrambled in the process.

Apply gap elimination step Interactions are added to the network layout as motifs, e.g. a signaling interaction will lay down five nodes at once in a specific pattern. This approach can create gaps in the columns of the layout. If this option is selected, the layout engine will eliminate any gaps in the columns, creating a tighter layout. The downside is that simple source-target motifs could get scrambled in the process.

Incremental layer compression This option influences how nodes are assigned to columns when they are added incrementally to an existing layout. If it is selected, excess horizontal space will be squeezed out the set of nodes being added. You should experiment with this setting to see if it improves the layout of your particular incremental addition.

Compress Inherited Regions Submodels are laid out by creating copies of the root network layout for each region present in the submodel. Since it is usually the case that these region copies omit network elements that are present in the root, the default algorithm applies a compression step to each region layout to squeeze out unused space. Unchecking this option will skip this step.

Layer Assignment Method The hierarchical layout algorithm assigns each node to an initial layer (each layer represents a column in the left-to-right hierarchy). However, if there are too many nodes in one layer, the layout could get too tall to be practical. This parameter specifies the method that will be used to make the final layer assignment: *Greedy Bounded Layering*, *Lazy Bounded Layering*, or *Simple Layer Bounding*. If working with a tall layout that needs to be bounded, play around to see what combination of method and layer size produces the best result.

Maximum Layer Size This setting is used in conjunction with the method specified above to bound the height of the layout. While low settings will make for a tight layout, the rectangular organization that results may not reveal the structure of the regulatory cascades that jumps out when high values are used. Instead of limiting this value too much, it may be better to compress the network vertically, after it is laid out, using **Tools->Compress Network**.

Link Optimization Passes As described above, the default setting is to do zero optimization/cleanup passes following link layout due to the possible expense of this step for large networks. However, this setting allows you to preset the number of passes.

Link Sliders The following four settings control the shape of the cost function that is used to determine when a link turns towards its target. This cost function tries to balance the trade-off between avoiding link crossings at the expense of having longer or more indirect links. Unfortunately, it can be hard to anticipate what the effects are going to be for a given change in these settings.

Link Crossings This setting assigns the weight that is given to penalize link crossings.

Link Crossing Sensitivity This setting specifies the width of the cost function that penalizes link crossings. For example, setting this to **Sharp** will cause the cost of link crossings to grow very rapidly as the number gets further from the ideal (i.e. no crossings).

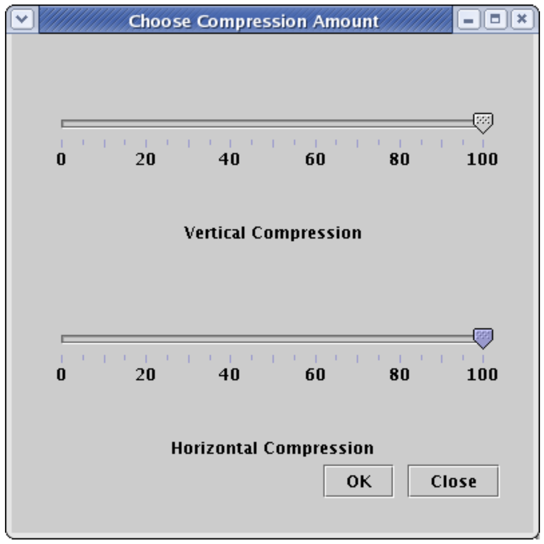
Link Overshoot This setting assigns the weight that is given to penalize overshooting or undershooting a link target.

Link Overshoot Sensitivity This setting specifies the width of the cost function that penalizes link overshoot. For example, setting this to **Sharp** will cause the cost of undershooting or overshooting the optimal turning point to grow very rapidly.

Other Layout Aids

Network Compression

It is frequently preferable to lay out a network initially with lots of extra space to accommodate crowded link paths and last-minute additions, and in fact the automatic layout engine does exactly that. The layout compression tool allows you to squeeze this space out when you are done. By choosing **Tools->Compress Network**, you can choose different settings for each dimension:



Setting a slider to zero will skip compression in the given dimension, while setting it to 100 will remove all available space.

The compression method completely ignores diagonal links, will maintain link orthogonality, will not change the relative positions of regions or network elements, and will generally not change link topologies, though in the current implementation, links between two different regions have to be redrawn to handle possible region size changes.

Given the conservative nature of the compression method, and because it maintains a relatively generous amount of padding around each element, it is likely that you can squeeze a layout down further by shifting elements around by hand. But it is a good place to start.

Network Expansion

If you wish to incrementally add new elements to an existing network layout, it is a good idea to expand the layout first. This gives you, or the layout engine, lots of space to work with to place new elements and route links through the network.

Selecting **Tools->Expand Network** brings up a dialog much like the compression one shown above. In this instance, a 100 setting will double the height or width of every row or column in the placement grid that qualifies as expandable.

Like the compression algorithm, the expansion algorithm ignores diagonal links, will maintain link orthogonality, will not change the relative positions of regions or network elements, and will generally not change link topologies, though in the current implementation, links between two regions have to be redrawn to handle possible region size changes.

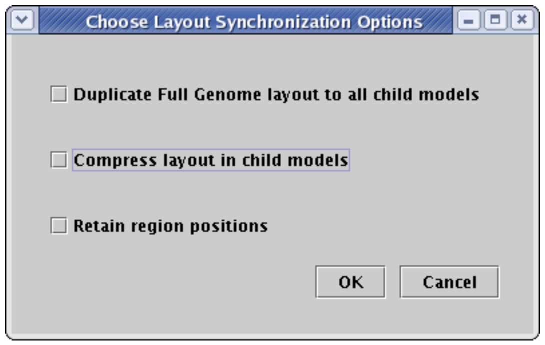
The expansion algorithm is *not* the inverse of the compression algorithm. In other words, a compression of x% followed by an expansion of x% (or vice versa) will not result in the same layout you started with.

Synchronize

The layout of the root network is completely independent from the layouts of the child networks, each of which has a layout that is independent of the others. This is necessary because each of these networks may be significantly different from each other. However, it is often useful to be able to synchronize the layouts of all these networks.

You can think of the network synchronization step as taking the root network layout and applying it to each child model layout by rubber stamping it for each region present in the child. The current relative positions of the regions in the child model can either be retained or recalculated. Links between regions will be drawn from scratch in either case, since those are unique for each child model and can change with the organization within each region.

To synchronize the layouts, choose **Tools->Synchronize All Layouts** from the main menu, which brings up this dialog:


Choose Layout Synchronization Options
Duplicate Full Genome layout to all child models
Compress layout in child models
Retain region positions
OK
Cancel

The dialog box will offer you either two or three options (the one shown above offers all three). In some cases, you are offered the choice to **Duplicate Full Genome layout to all child models**. That choice will only be active if every submodel of the root network has at most one copy of every root element. In that case, the root layout can be used as-is by all the child models.

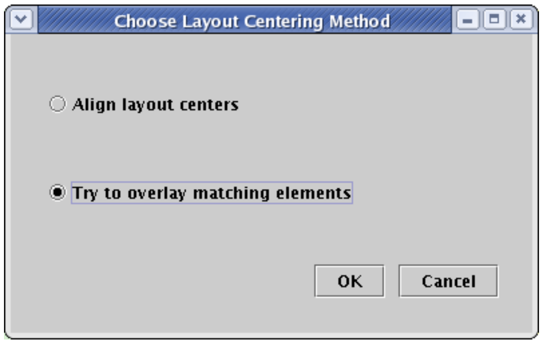
If the direct copy option is unavailable, or if you choose not to select it, then the other two options are made available. The first, **Compress layout in child models**, runs a separate network compression step on each fraction of the root network that is propagated down into a given region of a submodel. So if the different regions of your submodels include only fragments of widely separated portions of the root network, this compression option squeezes out lots of wasted space while maintaining the same overall layout within each region.

The remaining option, **Retain region positions**, allows you to maintain the existing locations of each region in the submodels. So, if you have created a particular arrangement of regions in your submodels that you want to maintain, you should select this option. The alternative is to have BioTapestry use its automatic region layout algorithm to choose how the regions are organized.

Align Centers

Since the layouts of the root network and all its child models are independent, you can end up with each network layout being offset from each other. It is often desirable to have the centers of all the network models match up, or to have elements shared between two or more network models coincident, The **Align Centers** tool helps with this process.

To align all the layouts, choose **Tools->Align Centers of Layouts** from the main menu, which brings up this dialog:


Choose Layout Centering Method
Align layout centers
Try to overlay matching elements
OK
Cancel

This dialog offers two mutually exclusive options for aligning all the different network views. You can choose to either **Align layout centers** or **Try to overlay matching elements**. The former is useful when your collection of layouts are rather dissimilar, and you just want to match the geometric centers of all the layouts. In contrast, the latter option is useful when the layouts are similar, and it is possible to overlay significantly identical portions of each layout.

Some Notes on Node and Link Color Assignments

When you draw a gene or a node, it is assigned a black color. Once you draw a link from a gene or some node types, BioTapestry assigns a color to both the source node and the link. This color assignment process does not make any effort to avoid unambiguous link crossings (i.e. where links from two different sources cross, but have the same color).

However, if you are creating a network from interaction tables, the layout engine does make an effort to assign node and link colors in a way that avoids ambiguous crossings. The engine has 32 colors to work with, and tries to distribute these colors to keep every link crossing unambiguous. If it cannot reach this goal, it will issue a warning message that will call out one (of possibly many) ambiguities that are present.

It is important to note that this color assignment process *only* occurs on the initial layout from an interaction list. Thus, even if you select **Discard previous layout** before re-laying out a network, BioTapestry still tries to keep node colors the same as before for to retain a measure of continuity, though it will attempt to reassign colors if needed to avoid ambiguous crossings. The only way to completely reassign colors is to erase the network first, but note that if you take this step, you will lose certain added properties of the nodes or links, like extra-long genes, that would otherwise be retained.

Since colors are not reassigned any other time during link relayouts, you may get link crossing ambiguity warnings anytime during a layout or link optimization operation. If this occurs, you can ignore it or you will need to use the ambiguity message to track down and manually change the source node and link color to remove the problem.

If there are simply too many link crossings to find a color assignment that avoids ambiguity, you can always fall back on using link branch bubbles. From the main menu, select **Edit->Set Display Options**, and set the **Special Linkage Branches** option to either **Filled** or **Filled with outline**. With these settings, link crossovers that do not have a bubble will be distinct from link branches.

Layout Failures

Sometimes the automatic layout engine has to give up trying to layout a link between two nodes. When this happens, it will display a warning message, and will create a simple, crude link tree with diagonal links between the source and target for each failed link.

Sometimes link failures will occur when the engine is trying to layout a network from scratch. This usually happens when you have a very large number of links inbound into a gene or node and it is difficult to find a clear path to the node, or if the number of free pads on the target node has been exhausted. If this happens, you will need to create a final layout manually.

Most often, however, link failures occur when you are working with an existing layout and you are trying to add new links, particularly if you are starting with a close-packed network with little extra space. If this is the case, it is usually best to expand the network a bit before incrementally adding new features. Also, small shifts to problematic node positions can frequently help the layout engine to find a successful route.

Useful Layout Tips

Remember that the default behavior for link layouts is that there are no optimization passes. It's always a good idea to run a few optimization passes manually following a link layout when you are ready to clean things up.

As the number of link optimization passes increases, you will see diminishing returns from each pass. Use the undo/redo function to compare the before and after link layouts see if anything is still changing.

If you are dragging genes or nodes around manually and plan on using the automatic link routing to fix up links, be careful not to place genes or nodes too closely together. If there is not enough space to work with when routing links, the layout engine will give up. It is better to leave lots of space, and then compress the network when you are done.

Along the same lines, if you have previously compressed the network (using **Tools->Compress Network**), it is wise to expand it (using **Tools->Expand Network**) before trying to add any new nodes incrementally, or doing link-only autolayout operations. Doing an expansion is particularly the best approach if the layout engine is giving up because it can't find a route though a crowd of links.

Building a Model Hierarchy Using Comma-Separated Value Input Files

BioTapestry can construct a complete model hierarchy from a comma-separated value (CSV) file exported by a spreadsheet program. This simple input format also provides an avenue to use BioTapestry as a visualization tool at the end of a computational pipeline, since it is simple to create a program that can write out data in the required CSV format.

Summary of CSV Input File Syntax

Any line in the CSV file where the first text field begins with a "#" is treated as a comment.

There are three basic types of instructions in the CSV input: *model*, *region*, and *interaction*. Furthermore, the interaction instructions are different for different types of interactions, and there are currently two subtypes for interaction instructions: *general* and *signal*.

Key to fonts used in tables

Bold indicates an item is a required keyword

Bold italics indicate and item comes from a limited set of possible keywords

Italics indicate an item should be customized with your own value

Model Instructions

Column Number	Contents
1	**model**
2	*model name*
3	*parent model name*

Notes:

1) The model hierarchy defined with model instructions must form a tree, i.e. one and only one model has no parent, and no cycles are permitted.

2) Model names must be unique.

3) Parent model names must reference a model created by another model instruction.

Region instructions

Column Number	Contents
1	**region**
2	*model name*
3	*region name*
4	*region abbreviation*

Notes:

1) The region name and abbreviation must be unique for a given model.

2) The top level model cannot have regions.

3) The model name must match a previously defined model.

4) If regions defined for a child model are not specified in the parent model, they are automatically included in the parent model.

5) Abbreviations must be no longer than 3 characters.

Interaction instructions (general subtype)

Column Number	Contents
1	**general**
2	*model name*
3	***source node type***
4	*source node name*
5	***target node type***
6	*target node name*
7	***interaction sign***
8	*source region abbreviation*
9	*target region abbreviation*

source node type = **gene** or **bare** or **box** or **bubble** or **intercel** or **slash** or **diamond**

target node type = **gene** or **bare** or **box** or **bubble** or **intercel** or **slash** or **diamond**

interaction sign = **positive** or **negative** or **neutral**

Notes:

1) The model name, source region abbreviation, and target region abbreviation must have been defined previously.

2) Interactions defined for the top model do not have source or target regions specified.

3) The type for a particular node name must be consistent across all instructions.

4) If an interaction defined for a child model is not specified in the parent model, it is added automatically.

Interaction instructions (signal subtype)

Column Number	Contents
1	**signal**
2	*model name*
3	**gene**
4	*signal source gene name*
5	**gene**
6	*target gene name*
7	**gene**
8	*mediated transcription factor name*
9	***signal type***
10	*source region abbreviation*
11	*target region abbreviation*

signal type = **promoteSig** or **repressSig** or **switchSig**

Notes:

1) The model name, source region abbreviation, and target region abbreviation must have been defined previously.

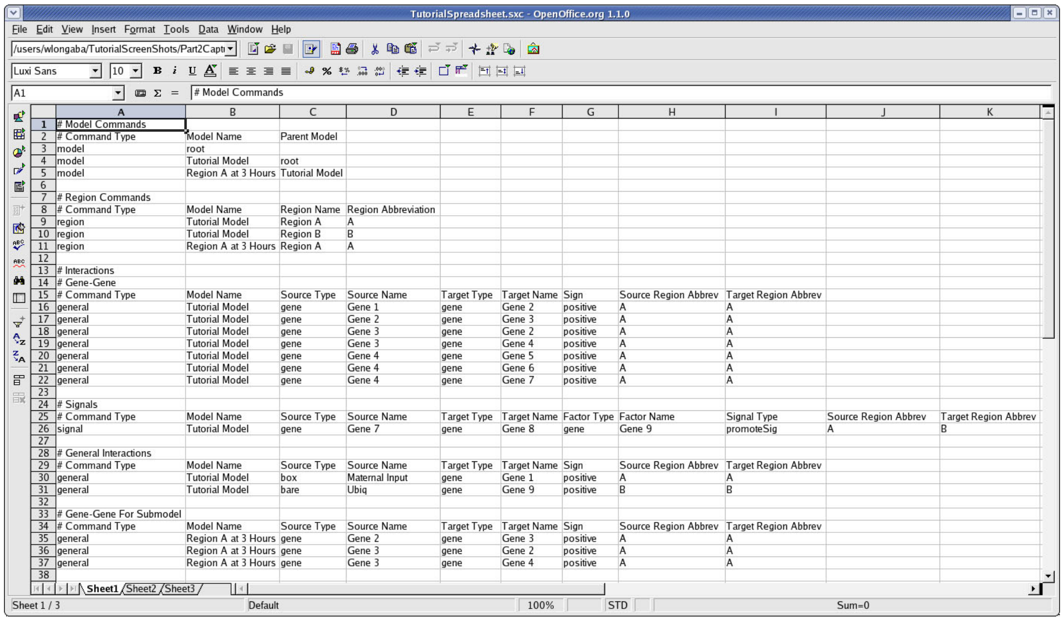
2) Interactions defined for the top model do not have source or target regions specified.

3) The type for a particular node name must be consistent across all instructions.

4) If an interaction defined for a child model is not specified in the parent model, it is added automatically.

How to Build from a CSV File

Using a spreadsheet program, create a file that matches the input syntax described above, then save it out by exporting to a comma-separated value (CSV format):



	A	B	C	D	E	F	G	H	I	J	K
1	# Model Commands										
2	# Command Type	Model Name	Parent Model								
3	model	root									
4	model	Tutorial Model	root								
5	model	Region A at 3 Hours	Tutorial Model								
6											
7	# Region Commands										
8	# Command Type	Model Name	Region Name	Region Abbreviation							
9	region	Tutorial Model	Region A	A							
10	region	Tutorial Model	Region B	B							
11	region	Region A at 3 Hours	Region A	A							
12											
13	# Interactions										
14	# Gene-Gene										
15	# Command Type	Model Name	Source Type	Source Name	Target Type	Target Name	Sign	Source Region Abbrev	Target Region Abbrev		
16	general	Tutorial Model	gene	Gene 1	gene	Gene 2	positive	A	A		
17	general	Tutorial Model	gene	Gene 2	gene	Gene 3	positive	A	A		
18	general	Tutorial Model	gene	Gene 3	gene	Gene 2	positive	A	A		
19	general	Tutorial Model	gene	Gene 3	gene	Gene 4	positive	A	A		
20	general	Tutorial Model	gene	Gene 4	gene	Gene 5	positive	A	A		
21	general	Tutorial Model	gene	Gene 4	gene	Gene 6	positive	A	A		
22	general	Tutorial Model	gene	Gene 4	gene	Gene 7	positive	A	A		
23											
24	# Signals										
25	# Command Type	Model Name	Source Type	Source Name	Target Type	Target Name	Factor Type	Factor Name	Signal Type	Source Region Abbrev	Target Region Abbrev
26	signal	Tutorial Model	gene	Gene 7	gene	Gene 8	gene	Gene 9	promoteSig	A	B
27											
28	# General Interactions										
29	# Command Type	Model Name	Source Type	Source Name	Target Type	Target Name	Sign	Source Region Abbrev	Target Region Abbrev		
30	general	Tutorial Model	box	Maternal Input	gene	Gene 1	positive	A	A		
31	general	Tutorial Model	bare	Ubiq	gene	Gene 9	positive	B	B		
32											
33	# Gene-Gene For Submodel										
34	# Command Type	Model Name	Source Type	Source Name	Target Type	Target Name	Sign	Source Region Abbrev	Target Region Abbrev		
35	general	Region A at 3 Hours	gene	Gene 2	gene	Gene 3	positive	A	A		
36	general	Region A at 3 Hours	gene	Gene 3	gene	Gene 2	positive	A	A		
37	general	Region A at 3 Hours	gene	Gene 3	gene	Gene 4	positive	A	A		
38											

TutorialSpreadsheet.sxc - OpenOffice.org 1.1.0
File Edit View Insert Format Tools Data Window Help
/users/wlongaba/TutorialScreenShots/Part2Capt
Luxi Sans
10
A1
# Model Commands
Sheet1 Sheet2 Sheet3
Sheet 1 / 3
Default
100%
STD
Sum=0

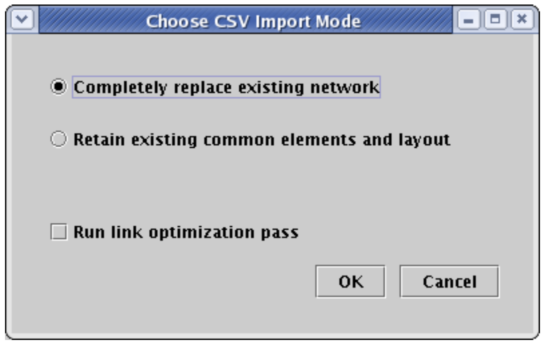
To import the CSV file, select **File->Import->Import Full Model Hierarchy from CSV...** from the main menu. You are then allowed to choose the method for integrating the CSV data into the existing network hierarchy:

Completely replace existing network: All existing models are dropped, and all layout information is discarded.

Retain existing common elements and layout: If you want to modify an existing network hierarchy with new data, you would probably choose this option, which will compare the existing network to the CSV input; both new models and new interactions will be added to the network hierarchy, and missing models and interactions will be deleted. The net effect of this operation is that the new hierarchy will exactly match the one specified in the CSV file, but existing layout for items retained by the new hierarchy will be kept. Please note that this does *not* mean that the CSV input will be added *on top* of what already exists.

The dialog also gives you the choice to request that a single pass of the link optimizer be run on all the imported networks. If you have set of large networks, it is likely that you would prefer to leave this unchecked, since link optimization can be expensive.

You can always run a link optimization step manually by selecting **Tools->Automatic Layout Tools->Run Single Link Optimization Pass**. If you have a small collection of small networks, selecting this option will provide a cleaner link layout with fewer artifacts. Once you have made your selection, click OK:



Choose CSV Import Mode

● Completely replace existing network

○ Retain existing common elements and layout

☐ Run link optimization pass

Command Line Arguments

When BioTapestry is executed using Java WebStart, you do not have any control over these arguments. However, they can be used if you are launching the program locally from a command line.

Set the Classpath

If BioTapestry is being invoked from the command line, you will need to set the classpath to include the required .jar files. For example, on Windows, where **%BTHOME%** defines the directory where the .jar files are located, you would need to issue the following command:

```
set CLASSPATH=%BTHOME%\sbioTapestry.jar;%BTHOME%\jh.jar
```

About Memory Requirements

When running the BioTapestry Editor, you need to specify a larger memory footprint than the default settings. In most cases, 512 megabytes is sufficient, so you add the argument **-Xmx512m**, as shown below. If you are working with a very large network and get `OutOfMemory` errors, this value may need to be increased.

Running the Program with Optional Arguments

The first version runs the editor, while the second runs the viewer:

Editor: `java -Xmx512m org.systemsbiology.biotapestry.app.EditorApplication`

Viewer: `java org.systemsbiology.biotapestry.app.ViewerApplication`

Arguments Summaries

These optional arguments can be provided after the class name (e.g. after org...EditorApplication as shown above).

-bigScreen

Will use very large fonts suited for QUXGA-W (3840x2400) displays.

-gaggle *speciesName*

Enables BioTapestry to connect with ISB Gaggle framework. Species name is used while communicating with the gaggle. This option is only valid for the editor application.

-plugInDir *directoryName*

Specify a directory to search for user-provided .jar files containing code for data display plugins. Each .jar file can hold one or more separate plugins. Each class bundled in the .jar file that implements the `ExternalDataDisplayPlugIn` interface should be listed in the following file bundled within the .jar:

META-INF/services/org.systemsbiology.biotapestry.plugin.ExternalDataDisplayPlugIn

For example, that file could contain the two lines:

```
org.systemsbiology.biotapestry.plugin.examples.HelloWorldPlugIn
org.systemsbiology.biotapestry.plugin.examples.HelloWebPlugIn
```

NOTE: This is intended for end-user provided plug-ins. If a developer wishes to bundle certain plugins with the application, it is better to register plugins in the `org.systemsbiology.biotapestry.plugin.plugInListing.xml` resource.

-resource *loadResource*
-remote *remoteURL*
-file *loadFile*

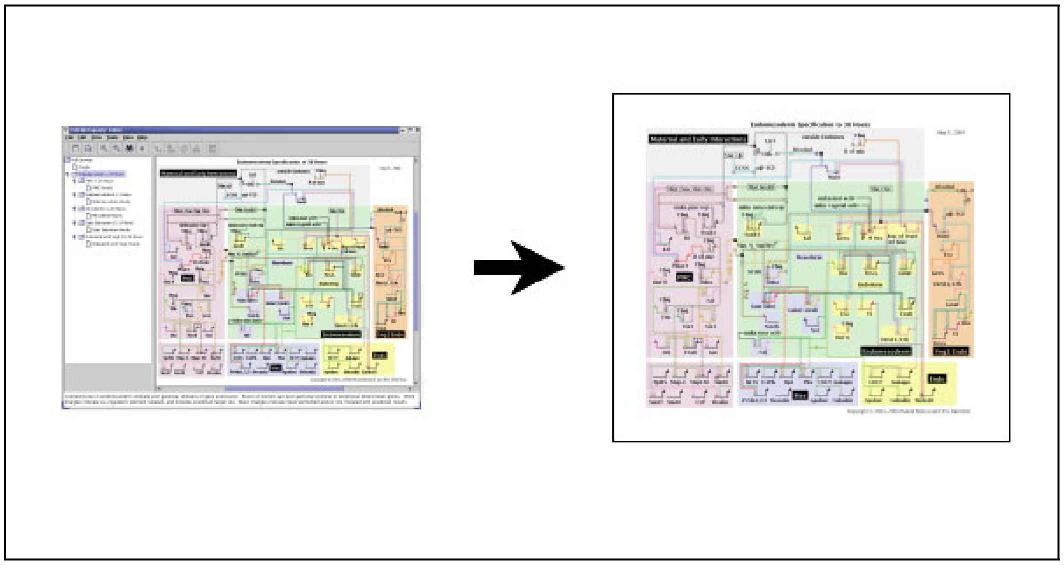
Specifies the .btp file to automatically load on startup. *Only one* of the three arguments should be present. The **-resource** option specifies a .btp file that resides inside a .jar file, and is useful for web-based viewing applications where the data is downloaded and cached on the user's machine. The **-remote** option allows you to specify a remote URL to fetch the .btp file. The **-file** option specifies a local file. The actual **-file** tag can be omitted if the file name is the last argument, but the other two tags must be provided if you want to use either of them.

BioTapestry Features And Tools

BioTapestry provides several ways to publish, share, and analyze the developed network model. Expanding these capabilities is one of the current areas of active development. Here is a list of some of the features that are not covered elsewhere in this guide.

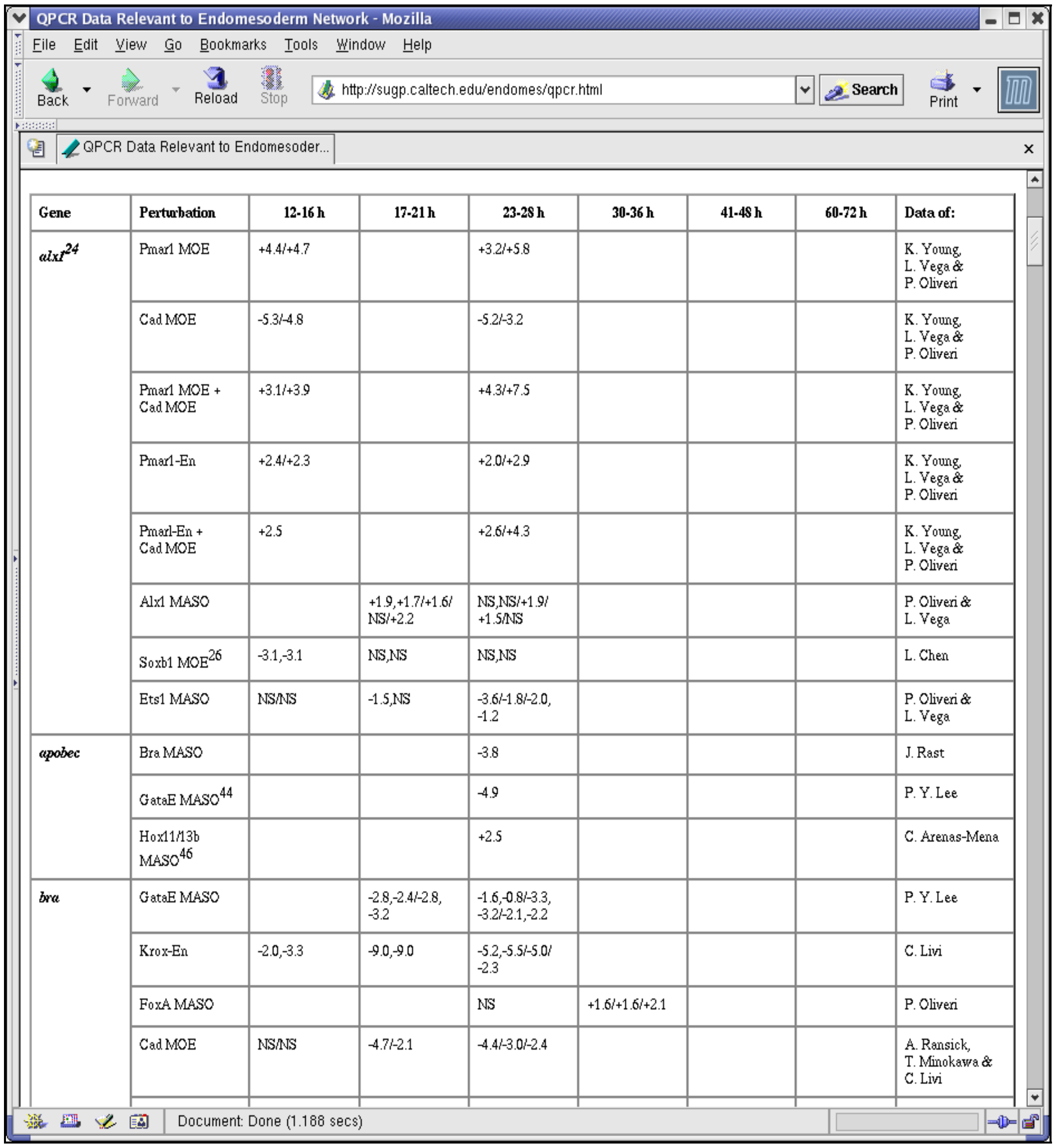
Image Print and Export

Both JPEG and PNG raster images of arbitrary size can be exported for any view of the network model. There is both a **simple image export** tool, and a more complete tool that lets you specify the image resolution and size for publication. The model can also be printed out directly:



QPCR Data HTML Export

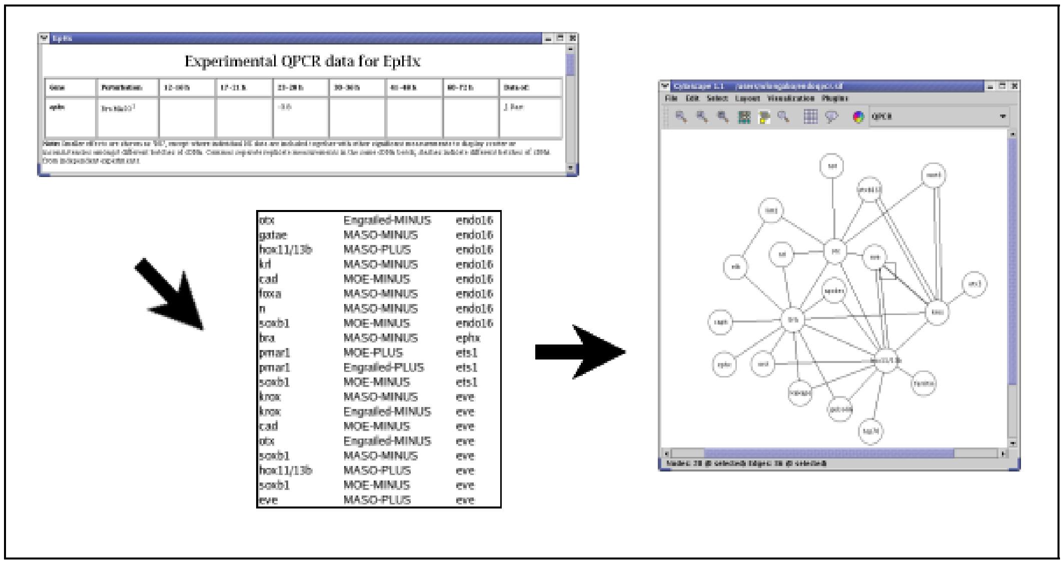
The tables of experimental QPCR data can be exported as HTML tables for publication on the web:


QPCR Data Relevant to Endomesoderm Network - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back
Forward
Reload
Stop
http://sugp.caltech.edu/endomes/qpcr.html
Search
Print
QPCR Data Relevant to Endomesoder...
Document: Done (1.188 secs)

Gene	Perturbation	12-16 h	17-21 h	23-28 h	30-36 h	41-48 h	60-72 h	Data of:
alx1[24]	Pmar1 MOE	+4.4/+4.7		+3.2/+5.8				K. Young, L. Vega & P. Oliveri
	Cad MOE	-5.3/-4.8		-5.2/-3.2				K. Young, L. Vega & P. Oliveri
	Pmar1 MOE + Cad MOE	+3.1/+3.9		+4.3/+7.5				K. Young, L. Vega & P. Oliveri
	Pmar1-En	+2.4/+2.3		+2.0/+2.9				K. Young, L. Vega & P. Oliveri
	Pmarl-En + Cad MOE	+2.5		+2.6/+4.3				K. Young, L. Vega & P. Oliveri
	Alx1 MASO		+1.9,+1.7/+1.6/ NS/+2.2	NS,NS/+1.9/ +1.5/NS				P. Oliveri & L. Vega
	Soxb1 MOE[26]	-3.1,-3.1	NS,NS	NS,NS				L. Chen
	Ets1 MASO	NS/NS	-1.5,NS	-3.6/-1.8/-2.0, -1.2				P. Oliveri & L. Vega
apobec	Bra MASO			-3.8				J. Rast
	GataE MASO[44]			-4.9				P. Y. Lee
	Hox11/13b MASO[46]			+2.5				C. Arenas-Mena
bra	GataE MASO		-2.8,-2.4/-2.8, -3.2	-1.6,-0.8/-3.3, -3.2/-2.1,-2.2				P. Y. Lee
	Krox-En	-2.0,-3.3	-9.0,-9.0	-5.2,-5.5/-5.0/ -2.3				C. Livi
	FoxA MASO			NS	+1.6/+1.6/+2.1			P. Oliveri
	Cad MOE	NS/NS	-4.7/-2.1	-4.4/-3.0/-2.4				A. Ransick, T. Minokawa & C. Livi

QPCR Data SIF Export

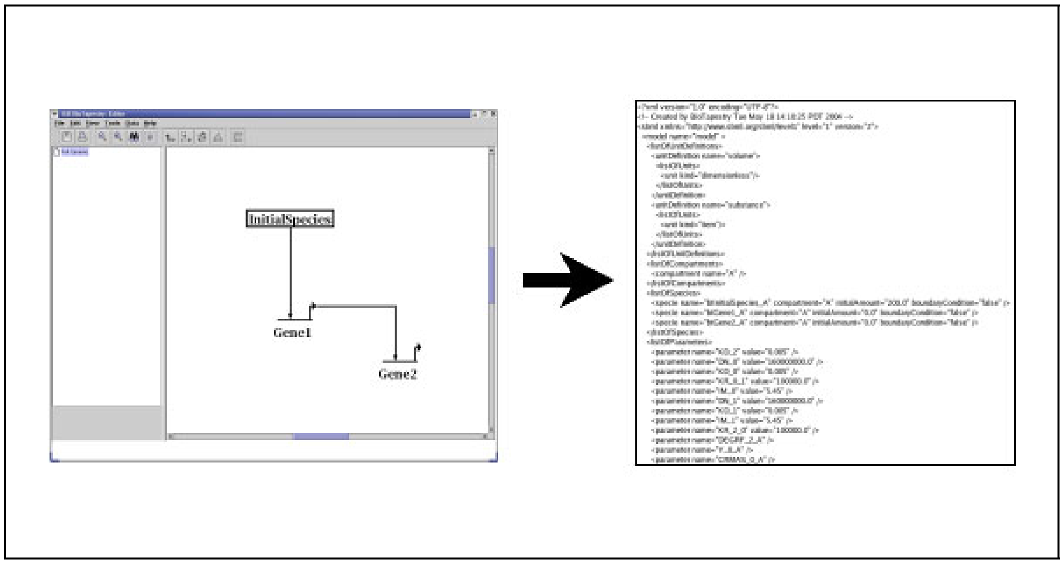
The interactions derived from the experimental QPCR data can be exported as a SIF file, allowing it to be viewed in Cytoscape:



otx	Engrailed-MINUS	endo16
gatae	MASO-MINUS	endo16
hox11/13b	MASO-PLUS	endo16
krl	MASO-MINUS	endo16
cad	MOE-MINUS	endo16
foxa	MASO-MINUS	endo16
n	MASO-MINUS	endo16
soxb1	MOE-MINUS	endo16
bra	MASO-MINUS	ephx
pmar1	MOE-PLUS	ets1
pmar1	Engrailed-PLUS	ets1
soxb1	MOE-MINUS	ets1
krox	MASO-MINUS	eve
krox	Engrailed-MINUS	eve
cad	MOE-MINUS	eve
otx	Engrailed-MINUS	eve
soxb1	MASO-MINUS	eve
hox11/13b	MASO-PLUS	eve
soxb1	MOE-MINUS	eve
eve	MASO-PLUS	eve

Export to SBML

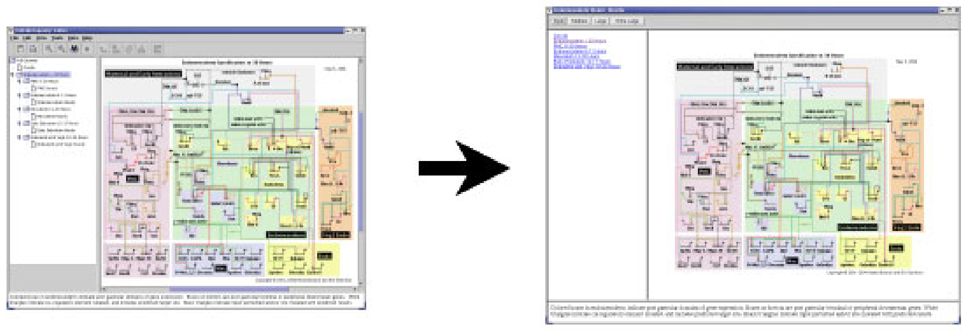
The network model can be exported as SBML, allowing it to be analyzed in other programs. (This ability is currently limited, but is under development):



```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Created by BioTapestry Tue May 18 14:18:25 PDT 2004 -->
<sbml xmlns="http://www.sbml.org/sbml/level1" level="1" version="2">
 <model name="model" >
  <listOfUnitDefinitions>
   <unitDefinition name="volume">
    <listOfUnits>
     <unit kind="dimensionless"/>
    </listOfUnits>
   </unitDefinition>
   <unitDefinition name="substance">
    <listOfUnits>
     <unit kind="item"/>
    </listOfUnits>
   </unitDefinition>
  </listOfUnitDefinitions>
  <listOfCompartments>
   <compartment name="A" />
  </listOfCompartments>
  <listOfSpecies>
   <specie name="InitialSpecies_A" compartment="A" initialAmount="200.0" boundaryCondition="false" />
   <specie name="btGene1_A" compartment="A" initialAmount="0.0" boundaryCondition="false" />
   <specie name="btGene2_A" compartment="A" initialAmount="0.0" boundaryCondition="false" />
  </listOfSpecies>
  <listOfParameters>
   <parameter name="KD_2" value="0.005" />
   <parameter name="DN_0" value="1600000000.0" />
   <parameter name="KD_0" value="0.005" />
   <parameter name="KR_0_1" value="100000.0" />
   <parameter name="M_0" value="5.45" />
   <parameter name="DN_1" value="1600000000.0" />
   <parameter name="KD_1" value="0.005" />
   <parameter name="M_1" value="5.45" />
   <parameter name="KR_2_0" value="100000.0" />
   <parameter name="DEGRF_2_A" />
   <parameter name="Y_0_A" />
   <parameter name="CRMAX_0_A" />
```

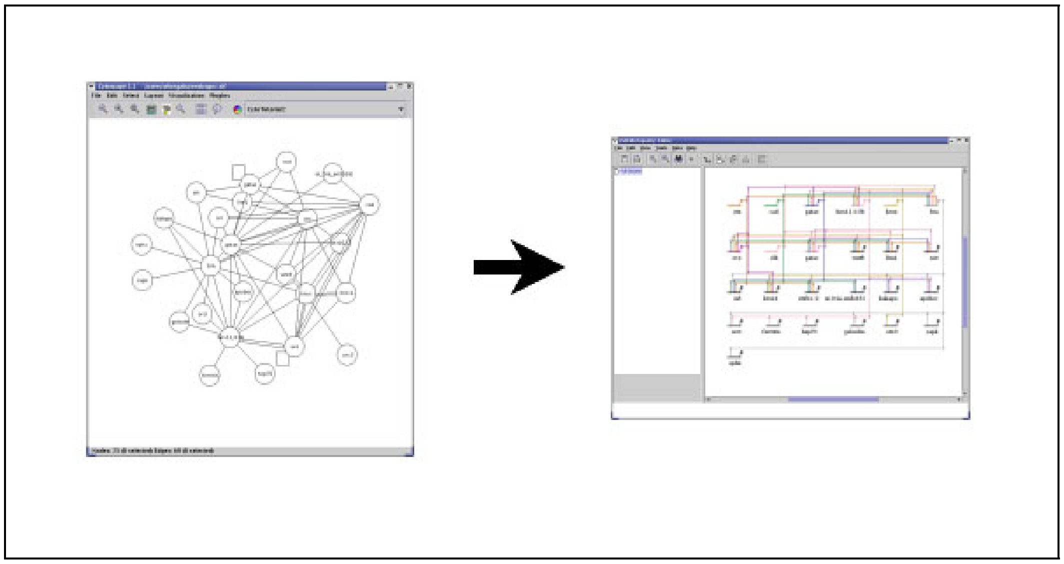
Web Export

The primary way to share models over the web is using the companion read-only BioTapestry Viewer program. However, BioTapestry can also produce a simple Javascript-only representation of the model that be accessed using a standard web browser that does not have Java WebStart installed:



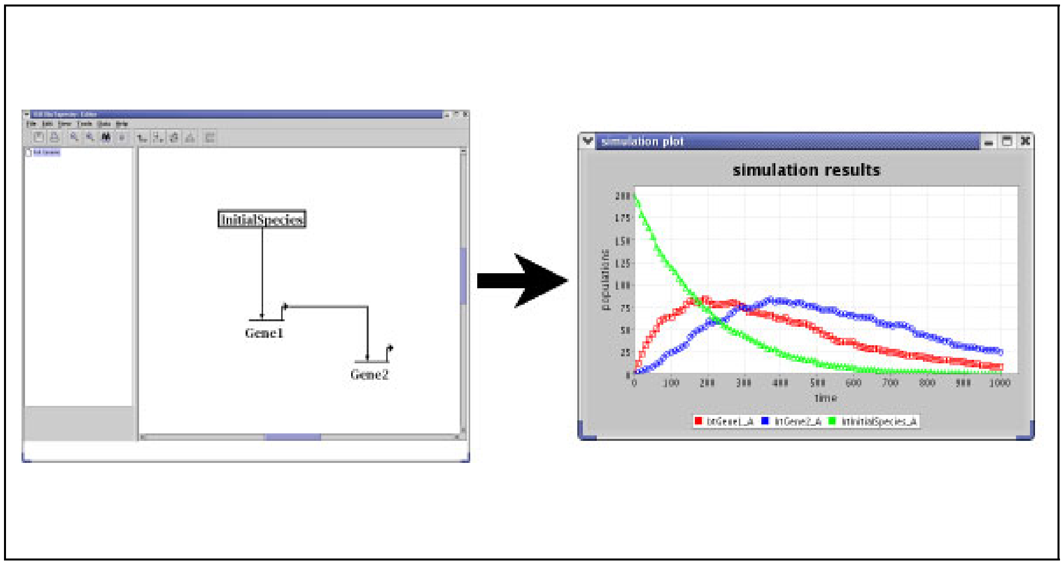
Import Network from SIF

A simple root-level network can be built automatically using a SIF file exported by Cytoscape:



Network Simulation

BioTapestry can launch network simulations using the Dizzy simulation engine. (This ability is currently limited, but is under development. It is not yet available in the BioTapestry version available through Java WebStart):



Building Network from QPCR Data

The raw QPCR data can be used to build a root-level model: