Published in final edited form as:
Biochim Biophys Acta. 2009 April ; 1789(4): 363–374. doi:10.1016/j.bbagrm.2008.07.014.

Visualization, documentation, analysis, and communication of large scale gene regulatory networks

William J.R. Longabaugh[1], **Eric H. Davidson**[2], and **Hamid Bolouri**[1,2,*]

[1] Institute for Systems Biology, Seattle WA 98103-8904, USA, W.J.R. Longabaugh: Institute for Systems Biology, 1441 North 34th Street, Seattle, WA 98103-8904, USA. Email: wlongabaugh@systemsbiology.org Phone: +1 206 732 1386 Fax: +1 206 732 1299

[2] California Institute of Technology, Pasadena CA 91125, USA, E.H. Davidson: 84 Alles, Division of Biology, California Institute of Technology, Pasadena CA 91125, USA. Email: davidson@caltech.edu Phone: +1 626 395 4937 Fax: +1 626 793 3047

Summary

Genetic regulatory networks (GRNs) are complex, large-scale, and spatially and temporally distributed. These characteristics impose challenging demands on computational GRN modeling tools, and there is a need for custom modeling tools. In this paper, we report on our ongoing development of BioTapestry, an open source, freely available computational tool designed specifically for GRN modeling. We also outline our future development plans, and give some examples of current applications of BioTapestry.

Introduction

As our understanding of genetic regulatory networks (GRNs) increases, ever more complex networks are studied. Ad-hoc ways of describing such networks (e.g. using generic drawing tools) are inefficient and inadequate, and there is an increasing need for specialized software.

In this paper, we present a software tool we have developed, called BioTapestry (http://www.BioTapestry.org/) [1], which has been designed from the ground-up to model GRNs. BioTapestry is free, open source, and runs on all popular computer platforms. Below, we illustrate some of the ways in which BioTapestry facilitates GRN modeling. Since BioTapestry is an active and ongoing project, we will also outline the requirements that will guide our future development efforts. We follow this with a roadmap on how to get started using BioTapestry, and finally give some examples of current applications of the software.

The need for a specialized GRN modeling tool

The architecture of a GRN arises directly from the DNA sequence of the genome, and a GRN model is directly testable by DNA manipulations. Thus, the representation of GRNs must be *genome oriented*, with specific emphasis placed on the predicted DNA inputs that form the

[*] Corresponding author: H. Bolouri: Institute for Systems Biology, 1441 North 34th Street, Seattle, WA 98103-8904, USA. hbolouri@systemsbiology.org Phone: +1 206 339 4852 Fax: +1 206 339 4852.

Publisher's Disclaimer: This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final citable form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

basis of the model. Furthermore, the GRN needs to be viewable at a number of different levels, from the whole, to the subcircuits, to the cis-regulatory DNA, and to the nucleotide sequence.

General-purpose network layout and presentation tools do not provide an appropriate level and style of abstraction for modeling GRNs. Many pathway modeling tools represent molecular interaction networks at the level of biochemical reactions. Because of the large number of reactions involved, representing GRNs as a set of biochemical reactions can result in overwhelmingly complex diagrams and obscure the regulatory architecture of GRNs. Moreover, the necessary biochemical data is rarely available to characterize such detailed views. On the other hand, overly abstract representations, such as those used in graph visualization software, lead to ambiguous network diagrams that convey little information.

Figure 1A shows a common graph-layout style GRN diagram. Compare this to the BioTapestry diagram in Figure 1B. The BioTapestry view immediately coveys a number of key concepts absent from the graph view. Firstly, *cis*-regulatory relationships are very easy to see and decipher in the BioTapestry view. Second, all nodes are not equal in BioTapestry layouts. BioTapestry uses automated layout templates to highlight regulatory relationships among the genes. For example in this figure, upstream regulators are placed near the top and to the left, while downstream genes are cascaded towards the right and bottom.

A key feature of GRNs is that a single gene will typically perform different regulatory interactions in different cells and at different times. A single static view of a GRN cannot convey the way a gene becomes part of different processes and functional modules in different cells and times. Figure 1B also shows how BioTapestry provides a hierarchical representation of GRNs which allow a user to track a GRN within a given group of cells over time, or to compare GRN state different between different cells at any given time.

Finally, BioTapestry is designed to facilitate the process of GRN model building and provides extensive support for network annotation and curation. We discuss these features in depth below.

GRN-specific representation

BioTapestry supports a symbolic representation of genes, their products, and their interactions, which emphasizes regulatory and experimentally-derived network features.

Representing GRNs at the cis-regulatory level

The most important concept to be communicated in GRN visualization is how the transcription of a gene is regulated by other genes in the network. This crucial information must be instantly recognizable from a cursory inspection of the network diagram. The representation of a gene, and in particular the *cis*-regulatory region of the gene, must be unique, structured, and organized in a fashion that stands out and is quickly understandable.

BioTapestry depicts a gene with the commonly used shorthand representation shown in Figure 2A. The key feature is an explicit schematic representation of the *cis*-regulatory modules of the gene. As shown, multiple binding sites for the same transcription factor, and multiple *cis*-regulatory modules can be depicted. Potentially important regulatory features, such as the spatial ordering of transcription factor binding sites on DNA, are also preserved. Furthermore, each regulatory input can be provided with a colored annotation tag for documentation purposes. For example, we often use colored diamond symbols to indicate the type of experimental evidence available for each binding site.

The above “cartoon” representation facilitates whole-network visualization of GRNs. To allow more in-depth evaluation, BioTapestry provides a data page for each network element (e.g. a gene or an interaction). These data pages can be customized by the user to display tables, figures and other data, or illustrations of the internal *cis*-regulatory logic of a gene. We plan to enhance BioTapestry in the future to permit multiple drill-down displays, and to make it easier to customize the data page.

Compact representation of off-DNA interactions

BioTapestry depicts off-DNA interactions as simple, compact and distinct symbols that quickly provide a sense of the general nature of the process, and its regulatory inputs and outputs, while hiding the details. In this way, complex processes such as signal transduction are modeled in terms of their regulatory role within the GRN of interest. This makes it much easier to understand the GRN at a glance, and allows uncluttered visualization of large-scale GRNs.

BioTapestry's collection of symbols for off-DNA actions and interactions are shown in Figure 2A. They are designed to provide enough information so that a viewer can “mentally fill in” the details from general knowledge. For example the canonical Wnt pathway may be communicated by a single labeled symbol. In general, pathways that do not include multiple regulatory inputs can be summarized by a single input-output symbol to avoid clutter. In cases where the details do need be accessible to the user, right-clicking on the symbol allows the user to view the same type of (user-customizable) pop-up data page containing tables, figures and other data, as is provided for genes.

Simplified representation of post-transcriptional processes

Post translation processes that are not differentially regulated within a GRN of interest are not represented explicitly in BioTapestry. The outputs of transcription factor genes are typically shown as direct inputs into the regulated gene targets, with an implicit understanding of what that simplification represents. Sometimes, there are post-transcriptional steps that are critical components of the regulatory behavior of the network, e.g. translation inhibition. In these cases, an explicit series of one or more linked off-DNA symbols can be inserted into, and replace, the simple direct input link (see Figure 2B). Similarly, if the gene creates multiple distinct products that are relevant to the regulatory function, the single gene output can be split into several tagged links, one for each product (Figure 2C). The same approach can be used to model the regulation of gene outputs via interactions with micro-RNAs, such as destruction of mRNA or interference with translation. Transcription of miRNAs is regulated in the same fashion as any other genes and can be modeled in BioTapestry in the same way as for any other gene. Regulation by miRNAs is modeled as post-transcriptional interaction, as illustrated in Figure 2D.

Maximizing GRN readability

BioTapestry uses a variety of strategies to facilitate the visualization of large numbers of genetic linkages:

- Links are not each drawn separately, but are bundled together and drawn as a group, as shown in Figure 3A. This is an extremely efficient approach that significantly cuts down on visual clutter that results when there are many genes in a network.
- Coloring is used to distinguish between adjacent and overlapping lines. BioTapestry automatically assigns a color from a palette of visually distinct colors to each link source, and then uses that same color for all of the source's outbound links. This makes it much easier to visually tie a distant link source to its corresponding link traces while inspecting the network (Figure 3B).

- Unique layout algorithms try to take advantage of the bundled link style to improve readability.
- Interactive tools to find link sources and targets, and pop-up tool tips that identify link source names, help for quick identification of network connectivity.
- "Branch bubbles" (as illustrated in Figure 3B) can be optionally activated to mark true link intersections, which can significantly improve readability and eliminate crossing ambiguities when the network grows large.

Although the outputs from a single source are typically rendered in a uniform fashion, BioTapestry allows the user to optionally highlight particular links by properties such as thickness, color, or line style. In addition, certain presentation properties of links, genes, and other nodes can be assigned to specific model properties. For example, the thickness of a link can be tied to the type of experimental evidence available. Such links are rendered accordingly even if the model is laid out differently (see Figure 3C).

Representing different network subsets and states

The same underlying GRN behaves differently in different cell types, spatial domains, and environmental conditions, and at different times. BioTapestry is designed to help the user to organize these varying views of the network state in a coherent fashion, while helping the user to understand how these views are derived from the single underlying GRN. As illustrated in Figure 4 (see also [1]), BioTapestry uses a three-level hierarchy to describe a GRN:

1. The View from the Genome (VfG) provides a summary of all inputs into each gene, regardless of when and where those inputs are relevant. Only one copy of each network element is shown.
2. The View from All nuclei (VfA) is derived from the VfG, and contains the interactions present in different regions over the entire time period of interest. Each region in a VfA is a subset of the VfG, and sub-networks may be duplicated in different regions.
3. Views from the Nucleus (VfN): Each VfN describes a specific state of the network at a particular time and place. Inactive portions of the network are indicated in gray, while the active elements are shown colored.

Each of these hierarchical views provides a different perspective on the GRN. A researcher can start their exploration of the network at any level, depending on the data available and the researcher's interests. For example, VfGs offer a natural perspective on each gene's full regulatory program within a GRN. However, to study functional motifs in the network (e.g. mutual exclusion), which are highly dependent on specific temporal and spatial conditions, VfN diagrams would be the most appropriate view.

BioTapestry makes it easy to create and organize GRN models using this approach. The hierarchical framework is general enough to be useful in many different ways. For example, one can use the lower levels of the hierarchy to depict variations in network behavior due to different experimental conditions. Alternately, submodels can be used to highlight network components discovered with particular experimental methods, or highlight subsets of genes or interactions that meet some significant selection criteria.

In BioTapestry, all the network models are automatically kept consistent across additions and deletions of network elements. For example, when a user inserts a node into an existing link in the root BioTapestry model, the program propagates the new node to all submodels that include an instance of the link. Since links may be laid out differently, and there may be multiple link copies spanning multiple regions in submodels, this can be a complex process and its automation greatly enhances model consistency and integrity. Along these same lines, we plan to add even more features that will help the user to easily propagate newly added network features to targeted regions and submodels.

Using the GRN model as an information portal

As we discussed above, network elements are simplified, abstract representations of complex subsystems, be they detailed *cis*-regulatory logic networks controlling a gene or complex off-DNA interactions such as lengthy signal transduction pathways. Often, it is useful to have access to more detailed information.

In BioTapestry, right-clicking on any gene or symbol in a network give the user an option to pop up a data display page. For genes, this page is typically configured to display raw perturbation data, the generic expression and interaction data tables that are used to drive the dynamic models, or arbitrary user-specified text. However, the page is customizable using small code plug-ins. In particular, it is easy to write a plug-in that displays data from a web server.

The plug-in approach allows for great flexibility in what data can be displayed, which is appropriate given the wide variety of information that may be appropriate to show. For example, what evidence is available to support the conclusion that a regulatory input is direct? What data have been used to determine that a gene expresses in a particular cell type and time? What are the exact cell type, genetic background and experimental conditions that a model is based on? Which particular member of a family does a gene or gene-product refer to? Potentially, all these types of information can be essential to fully understanding a model and its limitations.

In future versions of BioTapestry, we plan to simplify the data page customization process so that users can install commonly used options without needing to create special code modules.

Visualizing network dynamics

It is much easier to understand a sequence of GRN state changes through animations and interactive manipulations. A key feature of the BioTapestry Editor is that the user can click on genes and linkages to query their properties (e.g. all target genes, experimental evidence, or alternative paths between two nodes). These interactive features make it much easier to see the underlying organization within a large and busy static view.

BioTapestry also provides strong support for representing network dynamics:

- Interactive time slider controls allow visualizations of sequences of GRN state changes. The visualizations are driven automatically from data tables, which can be built from either experimental data or simulation runs.
- The user can create visualization sequence *paths*, each of which is a series of network views that can be easily traversed in the specified order. This allows the user to smoothly and quickly follow the state of a GRN through a sequence that may not be contiguous in the model hierarchy (see Figure 5).
- Individual static GRN views can show intermediate levels of gene expression and link activity (not just binary on/off, fully colored or completely grey) in a variety of easy to understand ways. For example, variable expression levels can be depicted using pie glyphs, color saturation, and line thickness. The user can pick and choose between the desired representations (see Figure 5).

Support for continuously variable expression levels in dynamic time-slider models is a planned future enhancement.

Making interactive GRN models available on the web

BioTapestry is written in Java, which is a freely available, cross-platform web technology. The program can be run as a web-based application using the Java Web Start facility. This means that a user with Java installed on their computer can click on a link on a web page, and the program will be downloaded to their computer and start to run automatically. Using this framework, a stripped-down “read-only” version of the program, called the BioTapestry Viewer, allows web-based viewing and interactive exploration of published GRN models. This feature enables a GRN model to act as an interactive and dynamic information portal for dissemination of research results and classroom teaching. The facility also encourages the development of community consensus models for widely-studied GRNs, shared over the web and built up by long-range collaboration. Since it is possible to tie the annotation data displayed for each network feature to a web page (using the plug-in facility), collaborative web technologies such as wikis can be used to enable community discussion and feedback.

Presenting the GRN model within its spatial and temporal context

BioTapestry allows images to be displayed alongside each network view in the model hierarchy. This facility can be used to indicate the cells of interest in the embryo at the appropriate developmental stage, or to indicate other spatial and contextual information using illustrative cartoons. In this way, images can be used to provide biological context for GRN models (Figure 6). This is particularly useful in systems involving multiple signaling events and changing cellular neighborhoods (e.g. development). When combined with the viewing path feature described above, the smooth progression of GRN states can be tracked using the images as well as the network model.

In future BioTapestry development, we plan to introduce 3D pictures and to integrate the image representation with the navigation functions.

Interacting with other computational tools

BioTapestry top-level networks can be exported to other software tools using the Systems Biology Markup Language (SBML) (http://sbml.org/) [2]. There are currently over 110 SBML-compatible software packages. BioTapestry also can import and export Cytoscape (http://cytoscape.org/) [3] interaction files. Furthermore, ongoing BioTapestry development aims to support the Gaggle framework (http://gaggle.systemsbiology.org/) [4]. Integration with the Gaggle will allow BioTapestry to interactively exchange network models and data with other tools.

A particularly popular feature of BioTapestry is that it can read entire hierarchical network descriptions from spreadsheets. Such spreadsheets could be generated manually, or by ad-hoc processing of experimental data, or automatically by computational data analysis pipelines. For example, Figure 7 shows a single subnetwork view of a very large dataset, from the Halobacterium EGRIN project (http://baliga.systemsbiology.net/egrin.php) [4], that was built using the BioTapestry spreadsheet import feature.

Data analysis and network construction

A common challenge in GRN modeling is distinguishing between direct and indirect linkages. To help meet this challenge, BioTapestry provides a display of alternative paths between a source and target gene (Figure 8A). The user can then inspect the supporting data to determine whether each linkage is direct. Tools like this help the researcher to build the model up from raw data and explore possible network architectures consistent with the data. We are currently working to enhance BioTapestry to provide more evidence visualization tools and a supporting analysis pipeline that can build a plausible network model from the perturbation and expression data.

Network exploration

BioTapestry now includes an enhanced search tool (Figure 8B) that allows the user to search for all nodes matching or partially matching a given name. It also allows the user to find all sources or targets of a given node, optionally selecting the relevant network segments. Another tool allows the user to select the sources or targets of a given link segment. Even more search enhancements are anticipated. For example, we plan to modify the search tool to work across multiple models as well as the current model. Another planned feature will allow the user to select all nodes and links within *n* hops of a selected subnetwork, thus making it possible to find the local neighborhoods of that subnetwork.

Making it easy to update and maintain the network

In general, a GRN model will never be definitive and complete. Over time, the model will be expanded, pruned, revised, and refined; interactions that were once thought to be direct turn out to be indirect, and vice versa. Modifying and maintaining increasingly complex models over an extended period of time is challenging, and BioTapestry provides features to handle these issues. As one example, it provides an incremental layout feature so that new network elements can be added while retaining most of an existing layout. Since this is an important long-term goal, we are continuing to add improvements that simplify creation and modification of the network. For example, recent BioTapestry enhancements allow model hierarchy trees to be duplicated; regions in a model can also be duplicated. Nodes can be inserted into links and automatically propagated to submodels.

Providing extensive layout support customized to GRNs

Whether the layout of complex GRN models is meaningful and understandable is a subjective judgment dependent on factors such as the user's points of interest, and focus. The commonplace algorithms for laying out network diagrams typically do not take biological considerations into account in their layout. BioTapestry layout algorithms address this challenge in the following ways:

- BioTapestry layout algorithms take into account the meaning of different symbol types, and use this information to organize nodes in an understandable way. For example, since genes are centrally important to understanding GRNs, they receive priority treatment in the layout process, with other symbols serving a supporting role (see Figure 3C). Furthermore, these small gene-centric blocks can be grouped into distinct units, and then these mid-scale units are laid out according to a high-level organizational strategy.
- BioTapestry layout algorithms now use different layout templates to organize GRNs in pre-configured ways. For example, one template specifies that regulatory genes are located in a different region of the diagram from downstream target genes. Another template also separates regulatory genes from downstream target genes, and also specifies that genes are positioned such that spatial location is indicative of time of expression onset.
- As was mentioned previously, the support of model hierarchies can make layout issues much more complex, since it is highly desirable to be able to maintain layout consistency among the various highly related but still unique views of the model. It is frequently important that the user be able to move between different representations of pieces of the network while still being able to tie that back to a mental model of

the network as a whole. BioTapestry currently uses a network layout subsetting strategy for propagating an overall layout to submodels such that strict geometric ordering is maintained across views while compressing out unused space.

- BioTapestry works to provide a variety of layout support tools to help a user who is drawing networks by hand, such as per-link auto layout, or global color assignment tools that can eliminate link crossing ambiguities.

Supporting multiple levels of abstraction within network views

The level of abstraction that we have used in BioTapestry so far has worked well for up to medium-sized networks, but as networks grow in size and complexity, new ways of organizing and thinking about network elements are needed.

The simplest way in which BioTapestry aids the understanding of GRNs is interactivity. A typical GRN presentation can be hard to understand when first viewed; it is only after interactively interrogating it and studying the various levels of the hierarchy that the network organization and functional features become apparent.

One of the ways in we hope we to facilitate understanding of large-scale GRNs is by introducing process diagrams as an additional level of representation within BioTapestry (see Figure 9A for an example). Another form of higher-level grouping, but typically dealing with smaller network chunks, is the identification of functional blocks [5] (for an example, see Figure 9B). Identification of functional blocks (e.g. feedback loops) also allows the user to view the GRN as a smaller set of interacting units, each with a clearly understood function.

These additional views, together with the network representations we have already discussed, imply a natural ordering of four levels of abstraction that are appropriate for looking at GRNs:

1. High-level process diagrams
2. Medium-level functional blocks
3. Fine-grained gene-centric view
4. Detailed network descriptions that model the logic embedded in *cis*-regulatory modules, and the specifics steps of off-DNA interactions

We plan to enhance BioTapestry to support these different representations in a manner that allows the user to view the network at the chosen level of detail, and to switch between these views as needed.

Getting started with BioTapestry

The BioTapestry Editor and associated tutorials are available freely at http://www.BioTapestry.org. The only prerequisite is that the freely-available Java Runtime Environment (JRE), from Sun Microsystems, is installed on your computer. This is commonly the case. If not, JRE can be installed easily by following instructions on the BioTapestry home page. With Java installed, clicking on the BioTapestry start link will cause the Java Web Start system to download the software to your computer and run it. This system allows users to be kept up-to-date with the latest version. When a user is not connected to the internet, they can still run the program from a desktop icon using the associated files saved automatically when the BioTapestry Java Web Start link was last accessed. Note that although the software is downloaded and maintained via the web, your data stays on your machine as locally saved files and is never uploaded to the server.

Also on the BioTapestry home page are a set of online tutorials that are designed to help you learn how to use the program by taking you step-by-step through simple examples that highlight important and common program operations. For example, BioTapestry supports several different ways of creating networks. Firstly, GRN models can be created manually. In this case, BioTapestry provides the user with an easy way to draw the network by hand, with fine-grained manual control over network placement and link layout. This method is taught in a *Quick Start* tutorial. At the other end of the spectrum, for large-scale regulatory networks based upon large amounts of data (e.g. from high-throughput experiments or computational analyses), BioTapestry is able to import large network descriptions from spreadsheet files, and automatically layout the large number of network elements in a coherent fashion. This method is described in a separate tutorial on *Building Networks from Comma-Separated Value Files*. Between these two extremes, BioTapestry allows users to specify networks interactively through a set of dialog boxes that guide the user. BioTapestry then automatically generates the network layout, thereby avoiding mundane layout tasks; this method is covered in a tutorial on *Building Networks from Interaction Tables*

In addition to these and other tutorials, there is an extensive online Frequently Asked Questions (FAQ) list that covers many topics in considerable depth.

Recent applications of BioTapestry

BioTapestry was initially developed to model the sea urchin endomesoderm specification networks [8]. A regularly updated BioTapestry viewer for this network is available at: http://sugp.caltech.edu/endomes/

It has since been used to model GRNs in a wide variety of organisms, including drosophila [9], arabidopsis [10], ciona [11], and yeast [12].

Also, an increasing number of projects are taking advantage of BioTapestry's ability to share interactive GRN models over the web. For example, an interactive model of mouse ventral neural tube specification [13] is available at: http://www.mcb.harvard.edu/McMahon/BioTapestry/

This model demonstrates how multiple VfA models can be created in BioTapestry to track a continuously varying set of developmental domains as development progresses.

The EGRIN (Environment and Gene Regulatory Influence Network) model recently developed for *Halobacterium salinarum* [4] provides an example of using BioTapestry's CSV input facilities and new auto layout algorithms to visualize large networks generated from computational analysis of high-throughput data. See (Figure 7), and: http://baliga.systemsbiology.net/egrin.php

In this case, BioTapestry's model hierarchy provides an excellent solution for showing the many different states of a genome-scale model.

The zebrafish developmental GRN presented in this issue [14] is an example of a BioTapestry model developed by merging data from public databases with additional local and 3rd party experimental observations. An interactive web model is available at: http://www.zebrafishGRNs.org/

Finally, an interactive web model of the mammalian T-cell developmental GRN [18], which is available at: http://www.its.caltech.edu/~tcellgrn/ is an example of a BioTapestry model developed by merging data from local and 3rd party experimental observations with additional public sources.

Looking ahead

More and more GRNs are being characterized every day. The resulting models are increasingly complex, and integrate very large volumes of experimental data. As the size and complexity of GRN models grows, four inherent capabilities of BioTapestry will prove increasingly essential.

Firstly, BioTapestry network diagrams present an integrated view of (i) the high-level architecture of the network, (ii) the *cis*-regulatory features of individual genes, and (iii) the supporting experimental evidence.

Second, BioTapestry's hierarchical views of GRNs highlight regulatory differences among cells (visualized in VfNs), as well as regulatory changes over time (visualized with ‘slide shows’), while at the same time emphasizing the relationship of dynamic GRN modules to genomic organization (visualized in VfAs and VfGs).

Third, as larger and larger networks are studied, the size and complexity of datasets is increasingly making their ad-hoc interpretation difficult and error-prone. BioTapestry supports a structured process for curating and translating experimental data into GRN models.

Fourth, our ongoing work to integrate process diagrams, and functional block representations into BioTapestry models facilitates the comparison of GRNs from different organisms (see for example [15,16,17]), allowing new insights into the functional and logical architecture of GRNs, and the fundamental principles underlying genetic control of cellular function.

Acknowledgments

BioTapestry development is supported by NIGMS grant GM061005.

References

1. Longabaugh WJR, Davidson EH, Bolouri H. Computational representation of developmental genetic regulatory networks. Dev Biol 2005;283:1–16. [PubMed: 15907831]
2. Hucka M, Finney A, Sauro HM, Bolouri H, Doyle JC, Kitano H, Arkin AP, Bornstein BJ, Bray D, Cornish-Bowden A, Cuellar AA, Dronov S, Gilles ED, Ginkel M, Gor V, Goryanin II, Hedley WJ, Hodgman TC, Hofmeyr JH, Hunter PJ, Juty NS, Kasberger JL, Kremling A, Kummer U, Le Novere N, Loew LM, Lucio D, Mendes P, Minch E, Mjolsness ED, Nakayama Y, Nelson MR, Nielsen PF, Sakurada T, Schaff JC, Shapiro BE, Shimizu TS, Spence HD, Stelling J, Takahashi K, Tomita M, Wagner J, Wang J. The Systems Biology Markup Language (SBML): A Medium for Representation and Exchange of Biochemical Network Models. Bioinformatics 2003;19(4):524–531. [PubMed: 12611808]
3. Shannon P, Markiel A, Ozier O, Baliga NS, Wang JT, Ramage D, Amin N, Schwikowski B, Ideker T. Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks. Genome Res 2003;13:2498–2504. [PubMed: 14597658]
4. Bonneau R, Facciotti MT, Reiss DJ, Schmid AK, Pan M, Kaur A, Thorsson V, Shannon P, Johnson MH, Bare JC, Longabaugh W, Vuthoori M, Whitehead K, Madar A, Suzuki L, Mori T, Chang D, DiRuggiero J, Johnson CH, Hood L, Baliga NS. A Predictive Model for Transcriptional Control of Physiology in a Free Living Cell. Cell 2007;131:1354–1365. [PubMed: 18160043]
5. Longabaugh W, Bolouri H. Understanding the Dynamic Behavior of Genetic Regulatory Networks by Functional Decomposition. Current Genomics 2006;7(6):333–341. [PubMed: 18079985]
6. Oliveri P, Tu Q, Davidson EH. Global Regulatory Logic for Specification of an Embryonic Cell Lineage. Proc Natl Acad Sci USA. in press
7. Oliveri P, Walton KD, Davidson EH, McClay DR. Repression of mesodermal fate by *foxa*, a key endoderm regulator of the sea urchin embryo. Development 2006;133:4173–4181. [PubMed: 17038513]

8. Davidson EH, Rast JP, Oliveri P, Ransick A, Calestani C, Yuh C, Minokawa T, Amore G, Hinman V, Arenas-Mena C, Otim O, Brown CT, Livi CB, Lee PY, Revilla R, Rust AG, Pan Z, Schilstra MJ, Clarke PJC, Arnone MI, Rowen L, Cameron RA, McClay DR, Hood L, Bolouri H. A Genomic Regulatory Network for Development. Science 2002;295(5560):1669–1678. [PubMed: 11872831]

9. Levine M, Davidson EH. Gene regulatory networks for development. Proc Natl Acad Sci USA 2005;102:4936–4942. [PubMed: 15788537]

10. Wellmer F, Alves-Ferreira M, Dubois A, Riechmann JL, Meyerowitz EM. Genome-Wide Analysis of Gene Expression during Early Arabidopsis Flower Development. PLoS Genetics 2006;2(7):e117. [PubMed: 16789830]

11. Stathopoulos A, Levine M. Genomic regulatory networks and animal development. Dev Cell 2005;9 (4):449–62. [PubMed: 16198288]

12. Smith JJ, Ramsey SA, Marelli M, Marzolf B, Hwang D, Saleem RA, Rachubinski RA, Aitchison JD. Transcriptional responses to fatty acid are coordinated by combinatorial control. Molecular Systems Biology 2007;3:115. [PubMed: 17551510]

13. Vokes SA, Ji H, McCuine S, Tenzen T, Giles S, Zhong S, Longabaugh WJR, Davidson EH, Wong WH, McMahon AP. Genomic characterization of Gli-activator targets in sonic hedgehog-mediated neural patterning. Development 2007;134:1977–1989. [PubMed: 17442700]

14. Chan TM, Longabaugh W, Bolouri H, Chen HL, Tseng WF, Chao CH, Jang TH, Lin YI, Hsu HI, Huang CB, Hung SC, Yuh CH. Developmental Gene Regulatory Networks of the Zebrafish Embryo. this issue

15. Oliveri P, Davidson EH. Built to Run, Not Fail. Science 2007;315(5818):1510–1511. [PubMed: 17363653]

16. Hinman VF, Davidson EH. Evolutionary plasticity of developmental gene regulatory network architecture. Proc Natl Acad Sci USA 2007;104(49):19404–19409. [PubMed: 18042699]

17. Davidson EH, Erwin DH. Gene Regulatory Networks and the Evolution of Animal Body Plans. Science 2006;311(5762):796–800. [PubMed: 16469913]

18. Georgescu C, Longabaugh W, Scripture-Adams D, David-Fung E, Yui M, Zarnegar M, Bolouri H, Rothenberg EV. A gene regulatory network armature for T lymphocyte specification. PNAS 2008;105(51):20100–20105. [PubMed: 19104054]

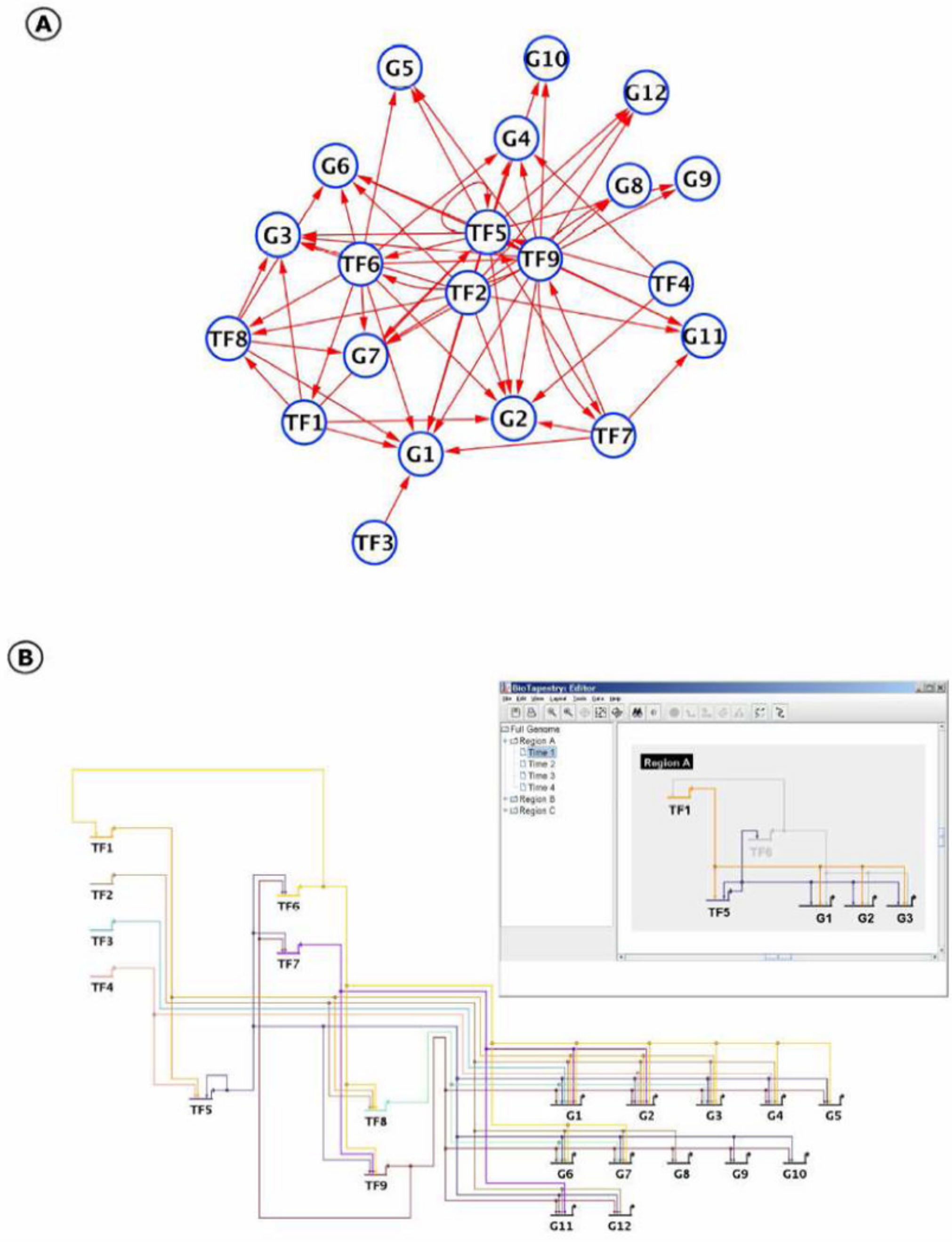


Figure 1.

Part A: A regulatory network presented using a common graph-layout style GRN diagram. Part B: The same network rendered using BioTapestry. Also shown (inset, upper right) is a screenshot displaying a subnetwork view of the GRN at a specific time in a specific region.

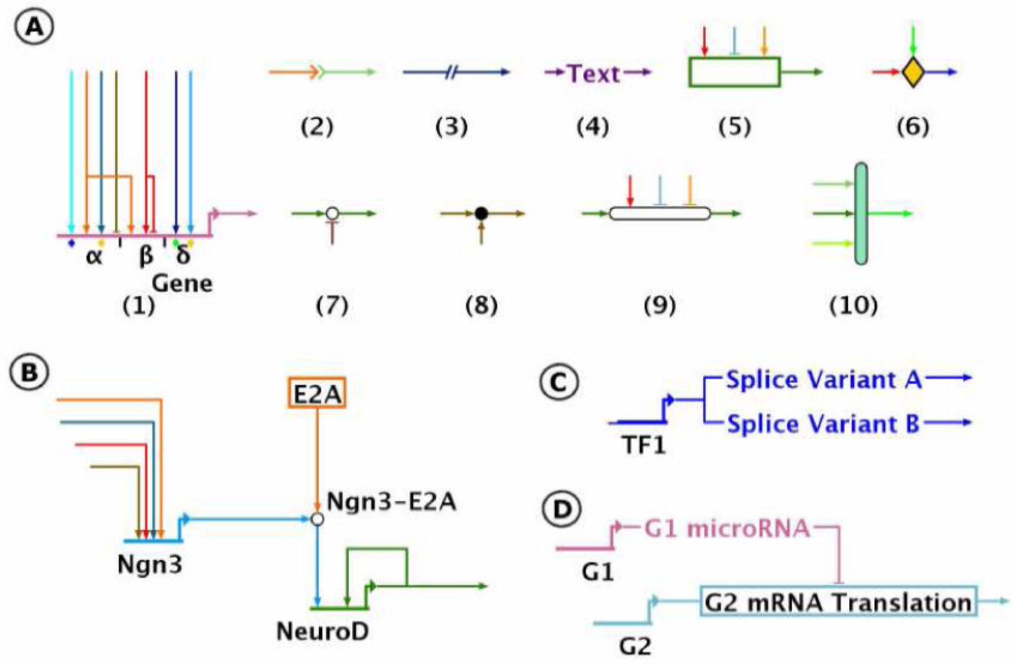


Figure 2.

BioTapestry's symbols and their use. Part A: The collection of symbols for genes and for representing off-DNA interactions and chemical species. Genes are shown in (1). The thick horizontal line is an abstract representation of the cis-regulatory region of the gene, which can be broken down into distinct labeled modules (here labeled α, β, and δ). All transcription factor inputs are shown as links terminating on the cis-regulatory region; arrowheads represent enhancers, while bars represent repressors. As the links are explicitly assigned to ordered terminals on the gene, the displayed ordering can be used as an abstract representation of the transcription factor binding sites in the regulatory region. Also, several inbound links from the same transcription factor can terminate in multiple distinct binding sites. Link inputs to the gene can be tagged to provide shorthand information; here, different colored diamonds are used to represent levels of experimental evidence. Other symbols are shown in (2) – (10). The program does not impose semantics on these symbols unless the user is building a model for simulation. However, by convention (2) represents a ligand–receptor interaction and (3) represents an indirect linkage. The plain text node (4) is often used without any inbound links to show nonspecific inputs (e.g. *Ubiquitous* or *Unknown activator*), and the box node (5) is used to show maternal factors. Note the link terminus on the center input to the box node in (5) represents a negative interaction. The diamond node (6) is often given a user-specific meaning. Bubble nodes (7) are typically used to represent protein-protein interactions and protein state changes (e.g., phosphorylation, degradation, or nuclearization). When colored black, as in (8), bubbles are typically used to represent the union of two equivalent paths (e.g. maternal and zygotically expressed protein products) onto the same targets. Most symbols can scaled to accommodate a large set of inputs, as shown for bubbles in (9) and (10). Part B: When there are added off-DNA steps between transcription and regulatory binding, beyond the canonical translation/transport that is represented by a direct link in BioTapestry, additional symbols are inserted in the link. Here, Ngn3 and E2A form a heterodimer, which then binds to NeuroD. Part C: If a gene creates multiple distinct products that are relevant to the regulatory function, the single gene output can be split into several tagged links, one for each product. Part D: Regulation by miRNAs is also modeled as post-transcriptional interaction, as illustrated.

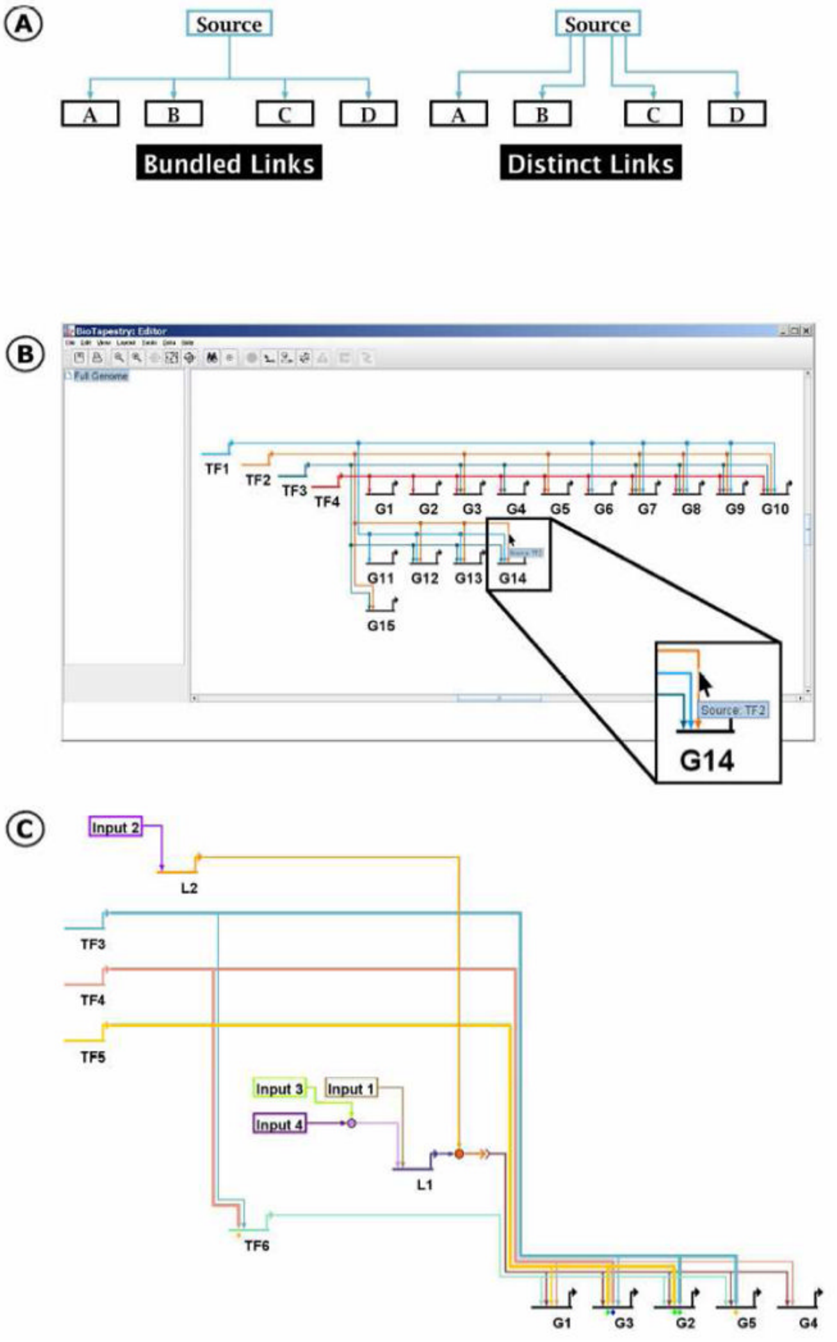


Figure 3.

Features of links in BioTapestry. Part A: By bundling together all the separate links from the same source, BioTapestry can efficiently represent links and cur down visual clutter. Part B: Color is an essential component of link appearance in BioTapestry. BioTapestry assigns a color to each link source, and then uses that same color for the outbound links. This approach makes it much easier to pick out the links associated with the source while inspecting the network. Even in cases when there are not enough colors to create an unambiguous rendering of the network, optionally activated “branch bubbles” (shown here) can remove crossing ambiguities. These tiny symbols can also make it much easier to visually pick out branches when quickly scanning along a link trace. Interactive features also help to track links. As shown in the enlarged inset, resting the cursor on a link reveals the link source. Part C: An example of automatically tying link presentation style to an underlying model property. In this case, link thickness is assigned to show the type of available evidence (as well as diamonds at the link terminii). Since this is automatic, the user does not have to be concerned with details of keeping the link rendering in sync with model changes, and can also rearrange the links without needing to reassign link thicknesses. This arrangement also shows how BioTapestry layout algorithms are gene-centric, with related off-DNA symbols arranged around each gene in a formal fashion based upon their role as inputs to, or outputs from, the gene. Furthermore, high-level strategies are used to group genes. Here, targets (genes with no regulatory role) are separated as a group from genes that are part of the regulatory network.

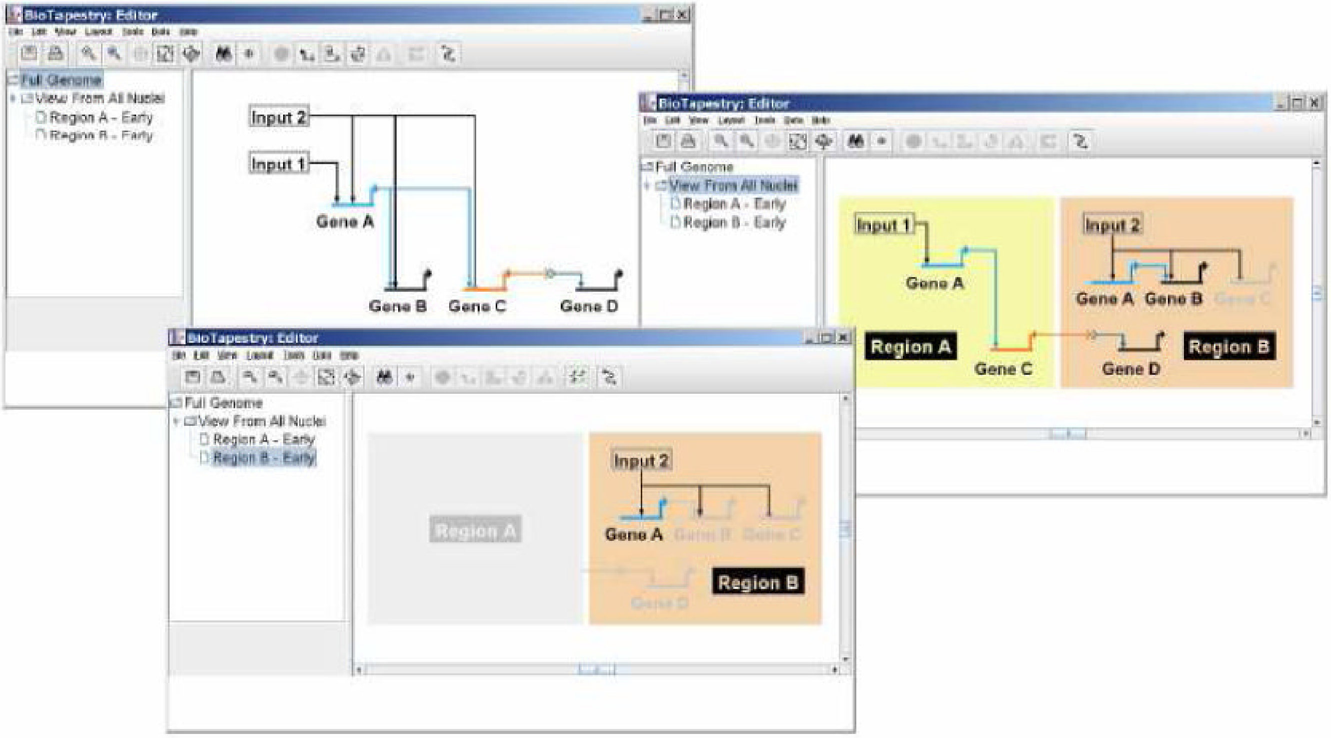


Figure 4.
Hierarchical models in BioTapestry. Clockwise from upper left: View from the Genome (VfG), View from all Nuclei (VfA), View from the Nucleus (VfN). See text for further discussion of each model level.

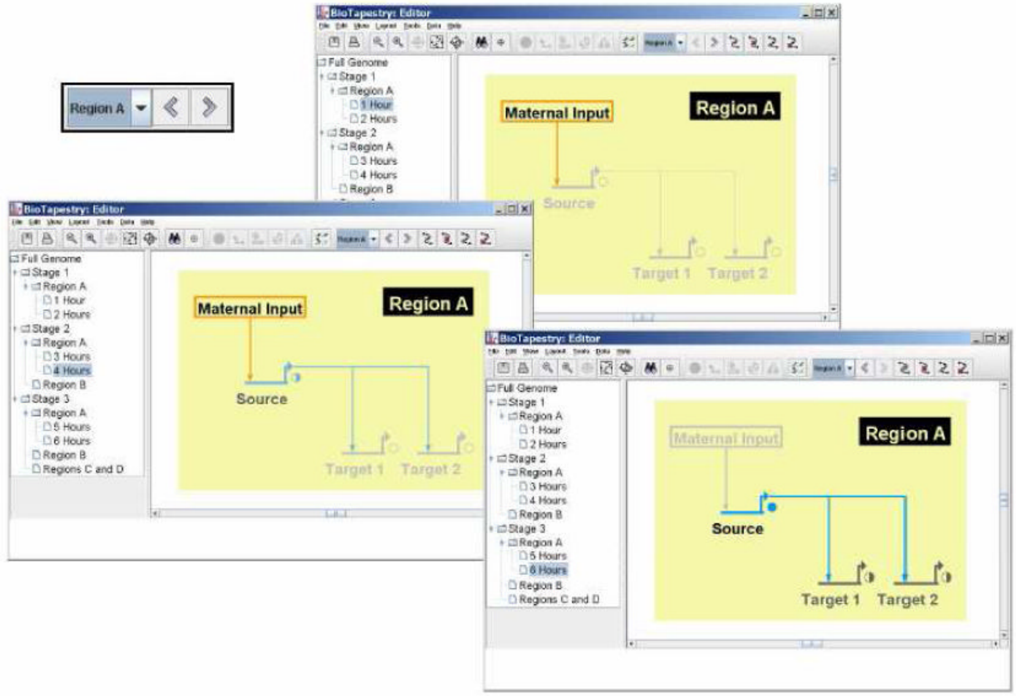


Figure 5.

User-defined model paths and intermediate expression levels are two new features of BioTapestry. The three screenshots show how a user-defined model viewing path can improve navigation. In many cases, following the temporal progress of a given region may involve navigation to submodels that are distributed throughout the model hierarchy, as shown here. Using a toolbar control (detail, upper left), the user can select a predefined named path, and then use the forward and back buttons to smoothly move along the path. In this example, the screenshots progress counterclockwise starting from the upper center, showing selected stops along a path tracking the progress of Region A over time. The screenshots also demonstrate intermediate expression levels. In previous versions of BioTapestry, genes and links were shown as either on or off. It is now possible (in static models) to show intermediate levels of normalized (0.0 to 1.0) gene expression and link activity using a variety of user-specified visualization methods. In this example, the source gene and its links go from a normalized expression level of 0.0 to 1.0 over the three illustrated time points in the series, while the target genes go from 0.0 to 0.5. All the different methods are shown here: 1) pie glyphs next to each gene show the normalized expression level; 2) the color is interpolated between the full-on color and the inactive light grey color (both for links and genes); and 3) the thickness of the link is varied.

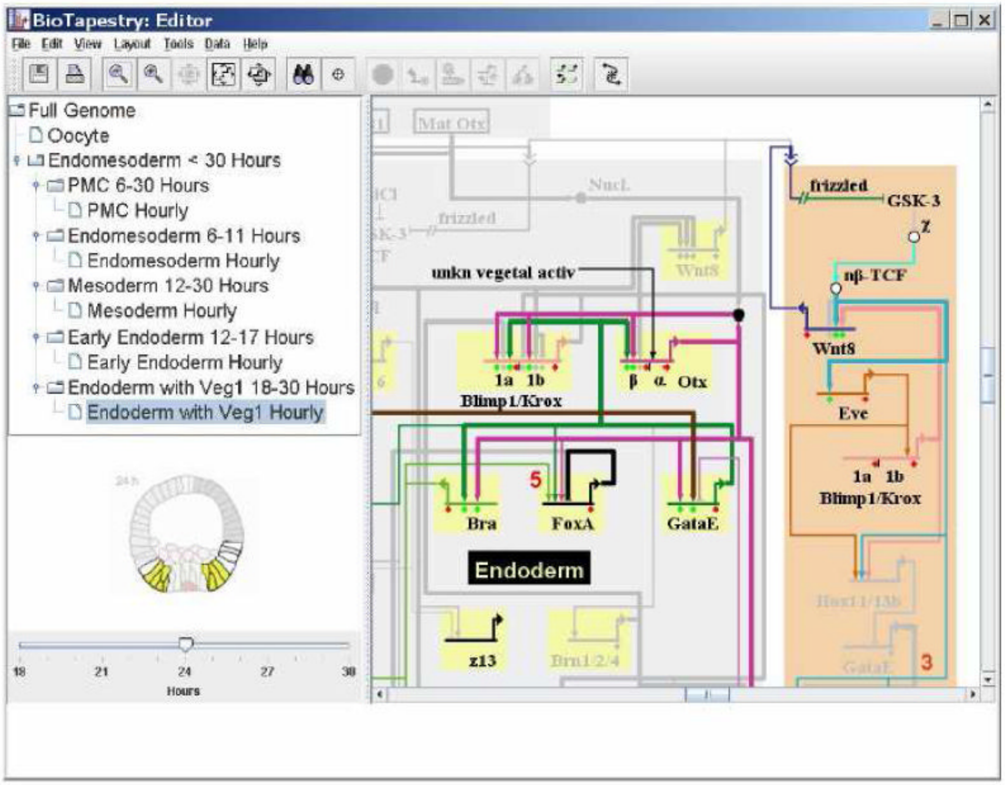


Figure 6.

Images can be displayed alongside each model, including each time point of a dynamic model, as shown here. The pictures provide the physical context for the abstract network depicted in the main window.

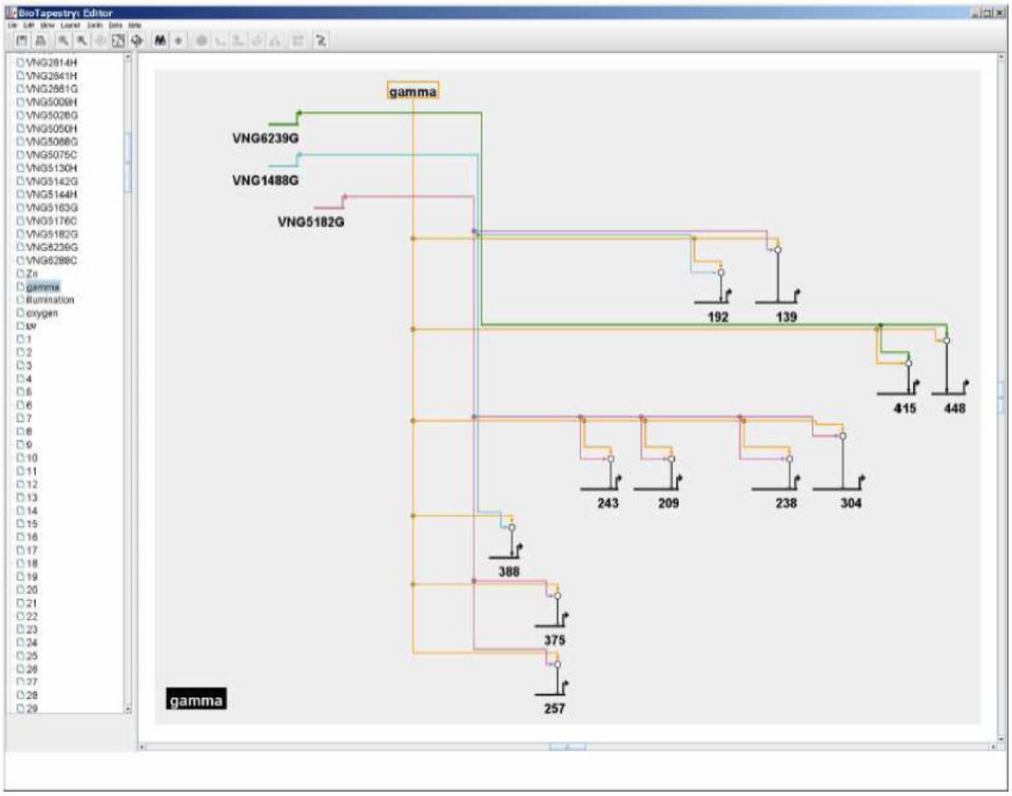
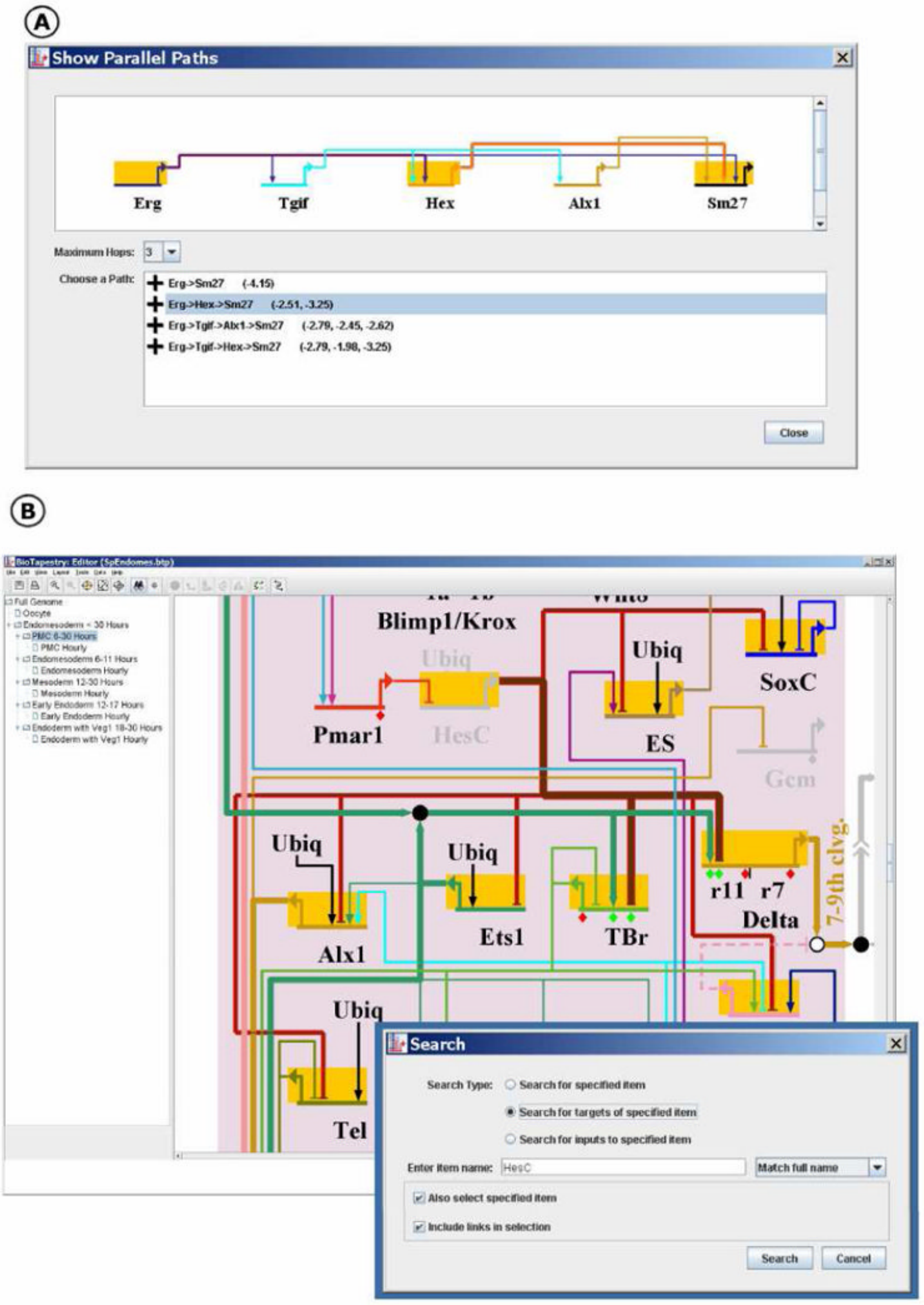


Figure 7.

The improved comma-separated value (CSV) file import and automatic layout capabilities of BioTapestry were used to generate a large network, with an extensive set of detailed submodels such as the one shown here, for the EGRIN (Environment and Gene Regulatory Influence Network) system for *Halobacterium salinarum* [4].



Maximum Hops: 3

Choose a Path:

- Erg->Sm27 (-4.15)
- Erg->Hex->Sm27 (-2.51, -3.25)
- Erg->Tgif->Alx1->Sm27 (-2.79, -2.45, -2.62)
- Erg->Tgif->Hex->Sm27 (-2.79, -1.98, -3.25)

Close

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

Search Type: ○ Search for specified item
● Search for targets of specified item
○ Search for inputs to specified item

Enter item name: HesC Match full name

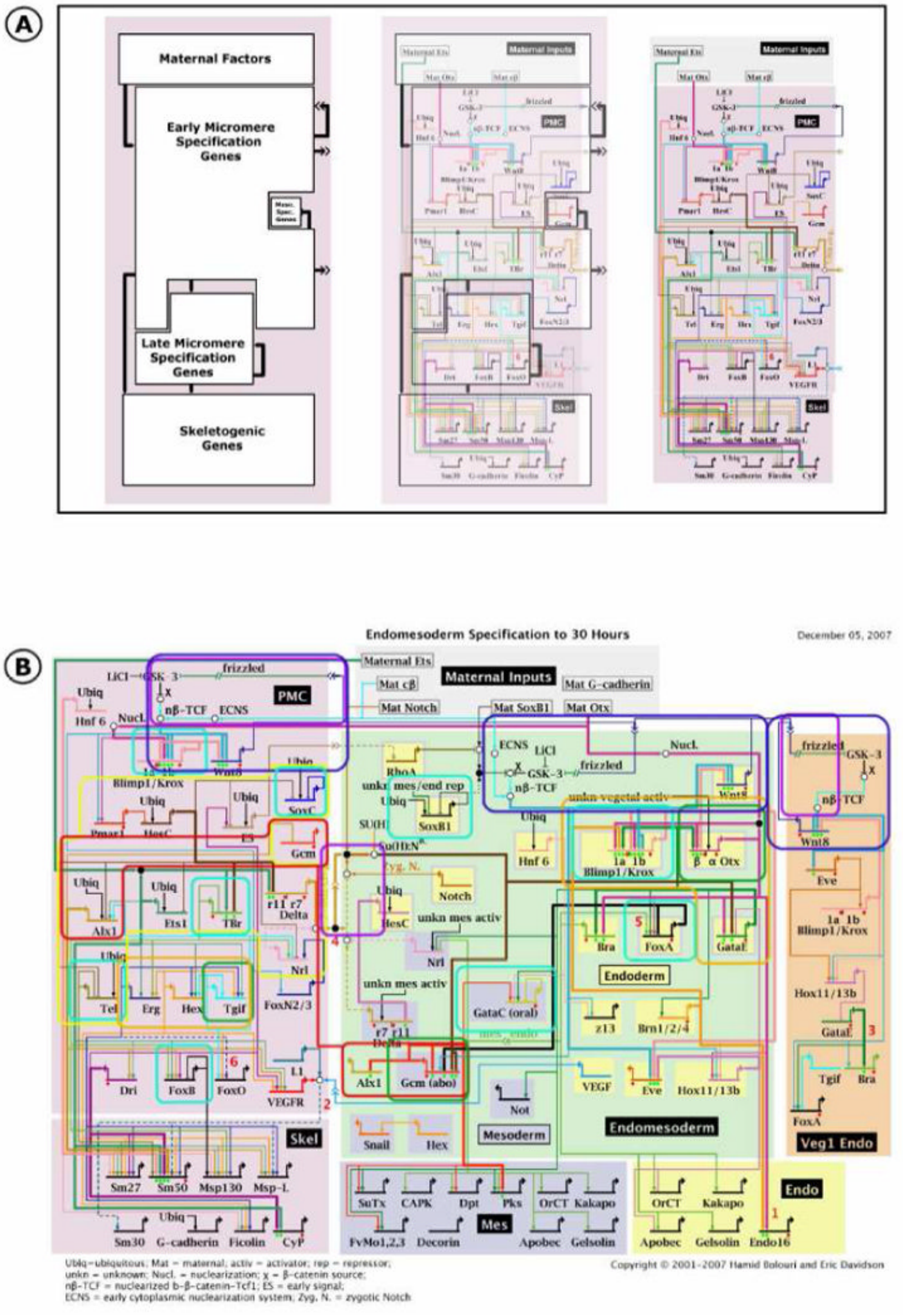
☑ Also select specified item

☑ Include links in selection

Search Cancel

Figure 8.

BioTapestry network exploration tools. Part A: BioTapestry provides a display to show alternative paths between a source and target gene. Tools such as these help the researcher to investigate possible alternative explanations of perturbation data while building the network from raw data. Part B: The recently enhanced BioTapestry search tool (dialog shown in inset) allows the user to select and zoom in on subnetworks. Here, the user has chosen to select all targets of *HesC*, including both the source gene and relevant link segments in the selection.



Endomesoderm Specification to 30 Hours

December 05, 2007

Ubiq=ubiquitous; Mat = maternal; activ = activator; rep = repressor;
unkn = unknown; Nucl. = nuclearization; χ = β-catenin source;
nβ-TCF = nuclearized b-β-catenin-Tcf1; ES = early signal;
ECNS = early cytoplasmic nuclearization system; Zyg. N. = zygotic Notch

Copyright © 2001-2007 Hamid Bolouri and Eric Davidson

Figure 9.

Proposed methods for introducing higher level abstractions into BioTapestry. Part A: A way to integrate process diagrams into BioTapestry, applied to the PMC domain of the sea urchin endomesoderm network. From left to right: a process diagram (see [6]) for the PMCs, an intermediate version that overlays the process diagram on the actual network components, and the actual network. Allowing the user to move smoothly between the two representations can significantly enhance his understanding of the network. Part B: The identification of functional blocks overlaid upon the concrete gene-and-link network can provide another conceptual framework for understanding the network. Shown here are putative functional building blocks in the sea urchin endomesoderm GRN; see [5]. Functional blocks are depicted by rounded bounding boxes of different color. Green: single-gene intra-cellular positive feedback latches. Orange: multi-gene intra-cellular positive feedback latches. Dark blue: inter-cellular positive feedback latch (the community effect) mediated by *wnt8* signaling. Cyan: instances of negative auto-regulation. *foxA* expression is oscillatory [7]. The expression patterns of the other auto-repressive genes are consistent with the 'single pulse' functional building block. Purple: signal mediated toggle switches mediated via β-catenin and TCF/LEF in Wnt signaling, and Su(H) in Notch signaling. Red: the *alx1-gcm* mutual exclusion operator. *Alx1* is on in the PMC domain and off in the mesoderm. *Gcm* is on in the mesoderm and off in the PMC. Yellow: the *pmar1* gradient detection/analogue to digital switch. Pmar1 represses the repressor *hesc*, which in turn represses *es*, *delta*, *nrl*, *alx1*, *tbr*, *ets1*, *tel*, and *soxc*.