A Compound Graph Layout Algorithm for Biological Pathways

Ugur Dogrusoz[1,2,⋆], Erhan Giral[1], Ahmet Cetintas[1],
Ali Civril[1], and Emek Demir[1]

[1] Center for Bioinformatics, Bilkent University, Ankara 06800, Turkey
ugur@cs.bilkent.edu.tr
[2] Tom Sawyer Software, Oakland, CA 94612, USA

Abstract. We present a new compound graph layout algorithm based on traditional force-directed layout scheme with extensions for nesting and other application-specific constraints. The algorithm has been successfully implemented within PATIKA, a pathway analysis tool for drawing complicated biological pathways with compartmental constraints and arbitrary nesting relations to represent molecular complexes and pathway abstractions. Experimental results show that execution times and quality of the produced drawings with respect to commonly accepted layout criteria and pathway drawing conventions are quite satisfactory.

1 Introduction

The notion of compound graphs has been used in the past to represent more complex types of relations or varying levels of abstractions in data [10, 8]. One such application is in bioinformatics; PATIKA (Pathway Analysis Tool for Integration and Knowledge Acquisition) is a software tool providing an integrated, multi-user environment for visualizing and manipulating network of cellular events [4].

There has been a great deal of work done on general graph layout [5] but considerably less on layout of compound graphs [14, 12, 2, 6], which has mostly focused on layout of hierarchical graphs.

There have been a few studies done specifically for layout of biological pathways as well, focusing on metabolic pathways [11, 1, 13]. Certain tools such as PATIKA enforce a more restricted ontology to represent signaling pathways whose underlying graph structure can be arbitrarily more complicated and irregular.

A layout algorithm for signaling pathways was proposed and implemented within PATIKA earlier [9]. However neither this algorithm nor any of the previously proposed ones address advanced pathway representations including nested drawings, intergraph relations, and application-specific constraints such as compartmental constraints at the same time. In this paper we describe a new algorithm for layout of compound pathway graphs.

2 Pathway Model

The structure of pathway graphs highly depend on the type of pathways (e.g., metabolic or signaling) and the model or ontology used to represent the biological

[⋆] To whom correspondence should be addressed.

J. Pach (Ed.): GD 2004, LNCS 3383, pp. 442–447, 2004.

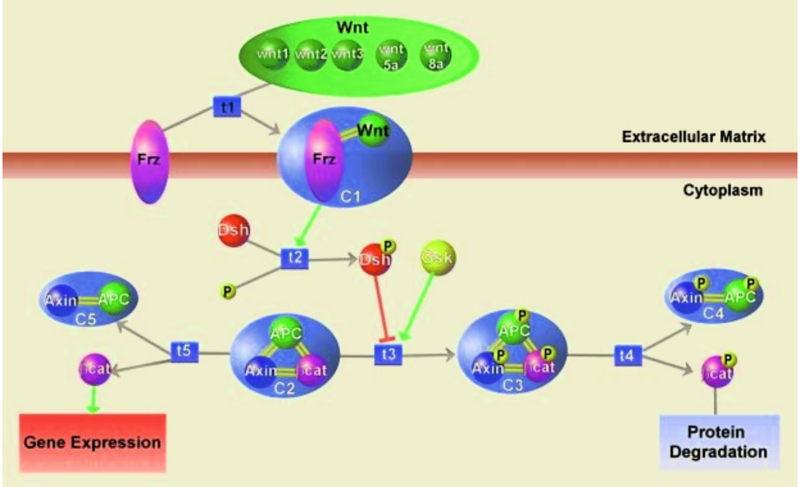

Fig. 1. Canonical wnt pathway represented by PATIKA ontology, including molecular complexes (e.g., C2) and various abstractions (e.g., Wnt and Protein Degradation).

phenomenon. We assume the ontology described in [3], which represents a cellular process in the form of a directed compound pathway graph (Figure 1).

3 Layout Algorithm

A force-directed layout algorithm with constraints to satisfy general drawing conventions in compound pathway graphs has been chosen. Basically, it is a virtual dynamic system in which nodes are assumed to be physical objects with a certain "electrical charge", connected via "springs" of a pre-specified desired length. Objects pull or repel each other depending on current lengths of any connected springs. In addition, relatively minor repulsion forces act on any pair of objects that are too close to each other to avoid node-to-node overlaps. Furthermore, each nested graph including the root of the nesting hierarchy is assumed to have a dynamic (with respect to the graph bounds) center of gravity. Thus the optimal layout is regarded as the state of this system in which total energy is minimal [7]. The following additions are made to this basic model (Figure 2):

- An expanded node and its associated nested graph are represented as a single entity, similar to a "cart" which can move freely in every direction. Multiple levels of nesting is modeled with smaller carts on top of larger ones.
- The nodes and edges of a nested graph are set in motion on this cart, confined within its bounds. Each cart is assumed to be surrounded by a material, elastic enough to adapt to the current bounds of the associated nested graph. Thus, as nodes of the nested graph are pushed out, expanding the nested graph, the expanded node adjusts its bounds accordingly.
- To avoid overlaps of variably sized nodes, desired edge lengths are calculated using parts of edges in between borders of end-nodes.
- Intergraph edges are treated specially; their desired lengths are set to be proportional to the nesting depth of the graphs their end nodes belong to.

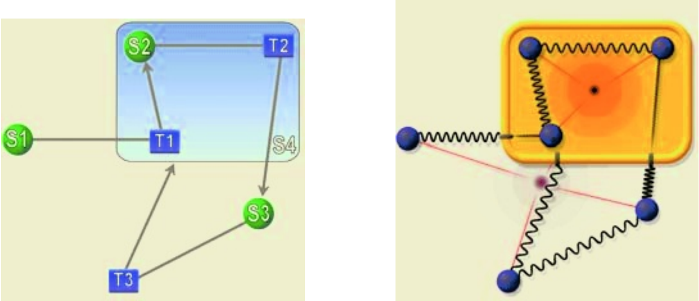

Fig. 2. Part of a sample compound pathway (left) and the corresponding physical model used by our algorithm (right).

We also apply *relativity constraint forces* or simply *relativity forces* on each substrate, product and effector states to position them properly around associated transition(s). The convention is to align substrates and products of a transition on opposite sides of the transition to form a certain flow direction. When calculating relativity forces, we first determine a flow, called *orientation*, for each transition by simply looking at current, relative positions of their associated substrates and products. Then each associated state of the transition is applied a relativity force to respect this orientation (Figure 3).

Another important constraint is due to cellular locations of biological nodes (compartments) represented by rectangular regions. The layout algorithm must keep each biological node within the bounds of the associated compartment and must enlarge or shrink it as required by the geometry of the pathway.

The algorithm is composed of three phases preceded by initialization:

Initialization: Initial node and compartment sizes, and threshold values for convergence are calculated as well as initial random positions of nodes. In addition, for efficiency and quality reasons parts of the given pathway that are

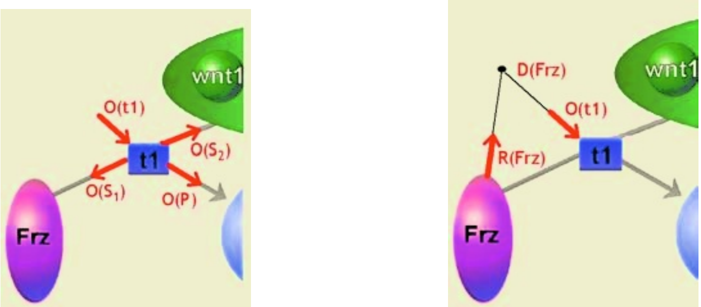

Fig. 3. An example of how the orientation of a transition is determined shown on transition t1 of Figure 1 (left) and used to calculate the relativity force on one of its substrates, Frz (right). O(t1), R(Frz), and D(Frz) respectively denote orientation of t1, relativity force on Frz due to t1, and desired location of Frz to obey this force, where magnitude of R(Frz) is equal to that of O(t1), and the distance of D(Frz) from t1 is equal to the desired edge length.

trees are temporarily removed. The remaining part of the pathway forms the "skeleton" of the pathway graph.

Phase 1: In this phase the skeleton graph is laid out using the spring embedder model described earlier but relativity and gravitational forces are disabled.

Phase 2: Trees reduced earlier on are introduced back level by level in this phase, also taking relativity and gravitational forces into account.

Phase 3: This phase is the stabilization phase where we "polish" the layout.

```
algorithm CompoundLayout()
call Initialization()
set phase to 1
if layout type is incremental then
    increment phase to 3
while phase $\leq$ 3 do
    set step to 1, error to 0
    while (step < maxIterCount(phase) and
            error > errorThreshold(phase)) or !allTreesGrown do
        call ApplySpringForces()
        call ApplyRepulsionForces()
        if phase $\neq$ 1 then
            call ApplyGravitationForces()
            call ApplyRelativityForces()
        call CalcNodePositionsAndSizes()
        call UpdateCompartmentBounds()
        if phase = 2 and !allTreesGrown and step % growStep = 0 then
            call GrowTreesOneLevel()
        increment step by 1
    increment phase by 1
```

A quick analysis reveals that the running time of layout is $O(k \cdot n^2)$ where n is the total number of nodes in the compound pathway, and k is the number of iterations required to reach an energy minimal state.

4 Implementation

The algorithm has been implemented within the Patika editor built on top of Tom Sawyer Software's GET for Java ver. 5.5. The results have been found satisfactory as far as the general graph drawing criteria such as number of crossings and total area are concerned. In addition, application-specific constraints such as compartmental constraints and relative positioning constraints seem to be highly satisfied. Figures 4 and 5 show sample pathway drawings produced.

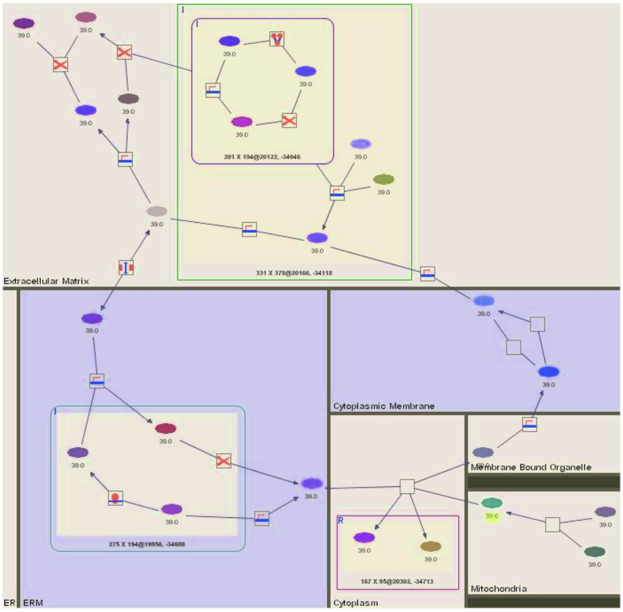

Fig. 4. Sample pathway from the PATIKA editor laid out by our algorithm.

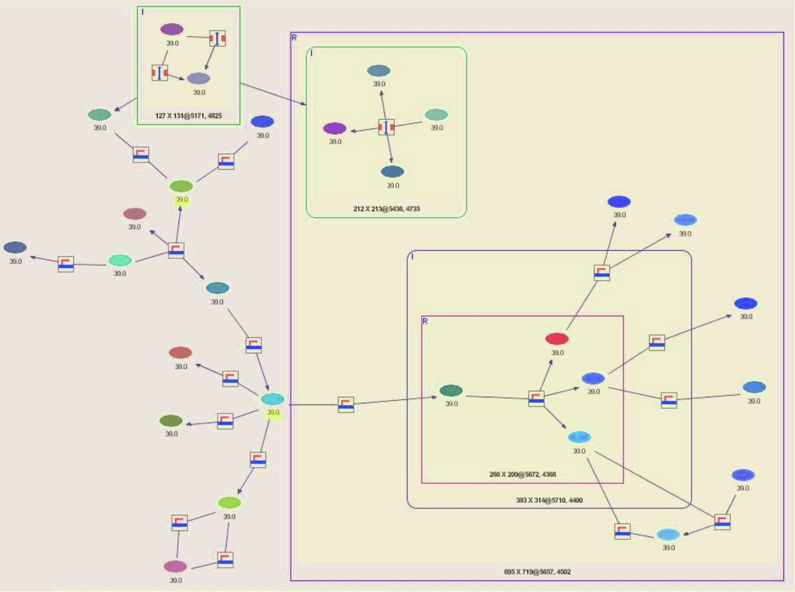

Fig. 5. Sample pathway from the PATIKA editor laid out by our algorithm.

From the theoretical analysis given earlier, a quadratic behavior of execution time versus number of nodes is expected, assuming k does not grow in the order of the graph size. The experiments validate this argument.

References

1. M. Y. Becker and I. Rojas. A graph layout algorithm for drawing metabolic pathways. *Bioinformatics*, 17:461–467, 2001.
2. F. Bertault. Force-directed algorithm that preserves edge crossing properties. In J. Kratochvil, editor, *GD '99*, volume 1731 of *Lecture Notes in Computer Science*, pages 351–358. Springer-Verlag, 1999.
3. E. Demir, O. Babur, U. Dogrusoz, A. Gursoy, A. Ayaz, G. Gulesir, G. Nisanci, and R. Cetin-Atalay. An ontology for collaborative construction and analysis of cellular pathways. *Bioinformatics*, 20(3):349–356, 2002.
4. E. Demir, O. Babur, U. Dogrusoz, A. Gursoy, G. Nisanci, R. Cetin-Atalay, and M. Ozturk. PATIKA: An integrated visual environment for collaborative construction and analysis of cellular pathways. *Bioinformatics*, 18(7):996–1003, 2002.
5. G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing, Algorithms for the Visualization of Graphs.* Prentice-Hall, 1999.
6. P. Eades, Q. Feng, and X. Lin. Straight-line drawing algorithms for hierarchical graphs and clustered graphs. In S. North, editor, *GD '96*, volume 1190 of *Lecture Notes in Computer Science*, pages 113–128. Springer-Verlag, 1997.
7. T. M. J. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software Practice and Experience*, 21(11):1129–1164, 1991.
8. K. Fukuda and T. Takagi. Knowledge representation of signal transduction pathways. *Bioinformatics*, 17(9):829–837, 2001.
9. B. Genc and U. Dogrusoz. A constrained, force-directed layout algorithm for biological pathways. In G. Liotta, editor, *GD '03*, volume 2912 of *LNCS*, pages 349–356, 2003.
10. K. Sugiyama and K. Misue. A Generic Compound Graph Visualizer/Manipulator: D-ABDUCTOR. In F. J. Brandenburg, editor, *GD '95*, volume 1027 of *Lecture Notes in Computer Science*, pages 500–503. Springer-Verlag, 1995.
11. P. D. Karp and S. Paley. Automated drawing of metabolic pathways. In *Third International Conference on Bioinformatics and Genome Research*, pages 225–238, Tallahassee, Florida, June 1994.
12. G. Sander. Layout of compound directed graphs. Technical Report A/03/96, University of Saarlandes, CS Dept., Saarbrücken, Germany, 1996.
13. F. Schreiber. High quality visualization of biochemical pathways in BioPath. *In Silico Biology*, 2(2):59–73, 2002.
14. K. Sugiyama and K. Misue. Visualization of structural information: Automatic drawing of compound digraphs. *IEEE Transactions on Systems, Man and Cybernetics*, 21(4):876–892, 1991.