Artificial Intelligence in Medicine 48 (2010) 107–117

Contents lists available at ScienceDirect

Artificial Intelligence in Medicine

journal homepage: www.elsevier.com/locate/aiim

Hierarchically organized layout for visualization of biochemical pathways

Jyh-Jong Tsay, Bo-Liang Wu [*], Yu-Sen Jeng

Department of Computer Science & Information Engineering, National Chung-Cheng University, 168, University Rd., Min-Hsiung, Chia-Yi 621, Taiwan, ROC

ARTICLE INFO

Article history:
Received 14 August 2008
Received in revised form 25 June 2009
Accepted 30 June 2009

Keywords:
Pathway layout
Forced-directed layout
Hierarchical layout
Pathway visualization

ABSTRACT

Objective: Many complex pathways are described as hierarchical structures in which a pathway is recursively partitioned into several sub-pathways, and organized hierarchically as a tree. The hierarchical structure provides a natural way to visualize the global structure of a complex pathway. However, none of the previous research on pathway visualization explores the hierarchical structures provided by many complex pathways. In this paper, we aim to develop algorithms that can take advantages of hierarchical structures, and give layouts that explore the global structures as well as local structures of pathways.

Methods: We present a new hierarchically organized layout algorithm to produce layouts for hierarchically organized pathways. Our algorithm first decomposes a complex pathway into sub-pathway groups along the hierarchical organization, and then partition each sub-pathway group into basic components. It then applies conventional layout algorithms, such as hierarchical layout and force-directed layout, to compute the layout of each basic component. Finally, component layouts are joined to form a final layout of the pathway. Our main contribution is the development of algorithms for decomposing pathways and joining layouts.

Results: Experiment shows that our algorithm is able to give comprehensible visualization for pathways with hierarchies, cycles as well as complex structures. It clearly renders the global component structures as well as the local structure in each component. In addition, it runs very fast, and gives better visualization for many examples from previous related research.

1. Introduction

Visualization of pathways always poses a challenge to bio-researchers. Though the idea is simple, drawing of complex pathways is not easy. In the past, pathways were drawn on papers. Recently, several pathway editors [1–3] have been developed. Such editors provide graphical interfaces, and graphical components that match pathway components. However, pathway data is increasingly large, and frequently updated. That makes drawing and maintaining of pathway data tedious and time-consuming, even we use graphical editors. A solution to the problem is to develop software tools that can automatically layout and visualize pathways.

One of the main problems in automatic pathway drawing is the layout problem. We can transform the relation between pathway components into a directed graph, and the drawing problem then becomes a graph layout problem. There are two well-known graph layout approaches: hierarchical layout [4–6] and forced directed layout [7,8]. However, both approaches can only handle simple pathway structures such as hierarchies and cycles. Recently, a number of approaches [9–12] have been proposed to layout pathways. In [9], Karp and Paley gave a pioneering work in which they pointed out that rather than searching for one single, all-purpose graph layout algorithm, different algorithms should be applied to parts of the pathway with different topologies. They devised an algorithm for drawing metabolic pathways which breaks the graph into cyclic, linear and tree-structured components and then applies different layout methods to each of these individually. Their algorithm has been implemented in MetaCyc [10] that will be used in this paper as a baseline to evaluate our approach. In [11], Becker and Rojas presented an algorithm that combines circular, hierarchical and force-directed layouts. Their approach is particularly designed for pathways that have a main cycle. In [12], Li and Kurata proposed a LK-grid layout algorithm that applies force-directed layout over grid points. They employed a cost function defined over connection structures in the graph. LK-grid layout is the first algorithm to produce clustered layouts in which nodes involved in the same functional module are clustered together. LK-grid layout has been integrated into CADLIVE [13] for automatic generation of pathway visualization. However, as observed from the examples in [13], LK-grid layout can induce

---

[*] Corresponding author. Tel.: +886 5 2720411x23123; fax: +886 5 2720859.
E-mail addresses: tsay@cs.ccu.edu.tw (J.-J. Tsay), wpl@cs.ccu.edu.tw (B.-L. Wu), cys92@cs.ccu.edu.tw (Y.-S. Jeng).

0933-3657/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.artmed.2009.06.002

large number of edge crossings, and local structures, such as hierarchies and cycles, are not clearly visualized. In [14], Kato et al. presented an improved grid layout algorithm that reduces the number of crossings, and incorporates subcomponent information. In this paper, we aim to develop new layout algorithms that produce clustered layouts, reduce edge crossings, and render global structures as well as local structures nicely.

Observe that many complex pathways are described as hierarchical structures in which a pathway is recursively partitioned into several sub-pathways, and organized hierarchically as a tree. The hierarchical structure provides a natural way to visualize the global structure of a complex pathway. However, none of previous algorithms explores the hierarchical structures provided by many complex pathways. In this paper, we develop a new layout algorithm, hierarchically organized layout (HOLY), for hierarchically organized complex pathways. HOLY first decomposes a complex pathway into sub-pathway groups along the hierarchical organization, and then partition each sub-pathway group into basic components. It then applies conventional layout algorithms, such as hierarchical layout and force-directed layout, to compute the layout of each basic component. Finally, component layouts are joined to form a final layout of the pathway. Our main contribution is the development of algorithms for decomposing pathways and joining layouts. Experiment shows that our approach can clearly render the global structure of complex pathways as well as the local structure in each sub-pathway, and gives better visualization for many examples from MetaCyc and CADLIVE. Furthermore, with the joining capability of our approach, one is able to visualize any collection of pathways, selected from pathway databases, to study their interactions.

The remainder of this paper is organized as follows. Section 2 reviews pathway visualization and conventional layout algorithms. Section 3 presents our hierarchically organized layout algorithm. Section 4 gives experimental results, and Section 5 concludes.

2. Pathway visualization

We follow the Petri Net representation of pathways in [15] that consists of two types of nodes: place nodes and transition nodes. As in Fig. 1, place nodes represented as circles denote compounds, and transition nodes represented as boxes denote reactions. Reaction nodes are labeled by Enzyme Commission numbers (EC numbers). A Petri Net becomes a directed graph if we treat both place nodes and transition nodes as the same. The problem of pathway layout thus becomes a graph layout problem.

Layout is a transformation from topology to geometry; that is, it generates coordinates for components. Forced-directed layout [7,8,16] is one of the widely used methods to produce a two-dimensional layout of a graph by doing simulation of physical systems. The basic idea is to model a graph as a physical system where the nodes are repulsive particles and the edges are attractive interactions. All the repulsive and attractive interactions are represented by an energy function. The algorithm first assigns a random coordinate to each node of the graph, and then iterates to update node coordinates by calculating all repulsive and attractive forces. Like a tied spring is going to gradually return to its natural form, an initialized graph will iteratively become a regular shape. A layout is determined when the forces drive the system to a steady state (a local minimum of energy). Force-directed layout can be applied to both directed graphs and undirected graphs. In general, there is no difference between them in that the force formulas between vertices are based on positions and distances, ignoring directions. Force-directed layout results in layouts with the following aesthetic properties [8]: distribute the vertices evenly in the frame, minimize edge crossings, make edge lengths uniform, reflect inherent symmetry and conform to the frame. In this paper, we will use force-directed layout for pathways that are mainly cycles.

Hierarchical layout [4–6,17] is another widely used method that features producing a layered downward diagram and highlighting the main direction or flow within a directed graph. Cyclic dependencies of nodes will be automatically detected and resolved. Nodes will be placed in hierarchically arranged layers. Additionally, the ordering of the nodes within each layer is chosen in such a way that the number of edge crossings is small. The algorithm aims to produce a layout that achieves the following goals [5,18].

- *Layering*: Each node has to be assigned a finite number of y-coordinates, and each arc has to be drawn from top to down.
- *Crossing minimization*: The order of node within a layer is acceptable when the node placement obtains a minimal number of arc crossings.
- *x-Coordinate adjustment*: Assign x-coordinate to each node obeying the fixed order and decrease the number of arc bends.

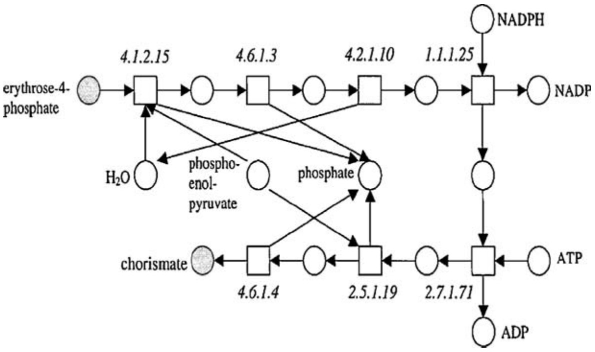

Fig. 1. Petri Net representation.

```
•  Data for reactions and compounds.
<global>
  <entry id="CIT" type="" nick="CIT" name=""></entry>

  …entries…

  <entry id="ACETYL-COA" type="" nick="benzoyl-acetyl-CoA"
   name="benzoyl-acetyl-CoA"></entry>

  <reaction id="CITSYN-RXN" name="Citrate (si)-synthase"
   ecnumber="2.3.3.1" left="CIT COA" right="OXALACETIC_ACID
   ACETYL-COA WATER" enzymeid=""></reaction>

  … reactions …

</global>

•  Data for pathways.
<global>
  <pathway id="TCA" type="GLYCOLYSIS-VARIANTS" name="TCA , TCA
   cycle -- aerobic respiration , tricarboxylic acid cycle ,
   citric acid cycle">
    <dlink id="d1" idfrom="MAL" idto="OXALACETIC_ACID"
   rxnid="MALATE-DEHYDROGENASE-(ACCEPTOR)-RXN"></dlink>

    … dlinks …

    <parent id="TCA_GLYOX-BYPASS"></parent>
    <variant id="P23-PWY PYRUVOX-PWY PWY-561 PYRUVDEHYD-
   PWY"></variant>
    <child id=""></child>
    <comment>The TCA pathway is …… captured as ATP, catalyzed
   by a multisubunit ATPase.</comment>
  </pathway>

  … pathways …

</global>
```

Fig. 2. The XML storage used in HOLY.