Algorithms for Various Biological Networks
BhartiTalukdar

Abstract: Biological networks are the networks which are used to represent different biological entities and relationship between the different entities. But due to the ongoing growth of knowledge in the life science their size and complexity is steadily increasing. For understanding biological networks several algorithms for lying out and graphically representing networks and network analysis results have been developed. However, current algorithms are specialized to particular layout styles and therefore various algorithms are required for representing different types of networks. This paper present a novel algorithm to visualize different biological networks and network analysis results in meaningful ways depending on network types and analysis outcome.

I. Background

Networks play a crucial role in biological analysis of organisms. They are used to represent processes existing in biological systems and to represent interactions and dependencies between biological entities such as genes, transcripts, proteins and metabolites. One large application area for network-centered analysis and visualization is Systems Biology, an increasingly important research field which aims at a comprehensive understanding and remodeling of the processes in living beings [1,2]. Due to the steady growth of knowledge in the life sciences such networks are increasingly large and complex. To tackle this complexity and help in analyzing and interpreting the complicated web of interactions meaningful visualizations of biological networks are crucial.

Since last few years methods for automatic network visualization have gained increased attention from the research community over recent years and various layout algorithms have been developed, e. g. [3-11]. Often standard layout methods such as force directed [12,13], layered [14,15] and circular [16] approaches are used to draw these networks. However, the direct use of standard layout methods is somewhat unsatisfactory since biological networks often have specialized layout requirements reflecting the drawing conventions historically used in manually laid out diagrams (which have been developed to better emphasize relevant biological relationships and concepts). This has led to the development of network- and application-specific layout algorithms, for example, for signal transduction maps [17,18], protein interaction networks [3,6], metabolic pathways [4,10,19] and protein-domain interaction networks [20]. Advanced solutions combine different layout styles (such as linear, circular and branching layouts) for sub-networks or use specific layouts for particular network parts such as cycles [7,10,21].

However, current approaches for the automatic visualization of biological networks have four major drawbacks resulting from the specialized nature of these algorithms:
1. Different kinds of biological networks (e. g. protein interaction or metabolic networks) have different layout conventions and this requires the implementation and sometimes development of specialized layout algorithms for each convention.
2. It is not easy to combine networks with different layout conventions in the one drawing since the layout algorithms use quite different approaches and so cannot be easily combined.
3. The user cannot tailor the standard layout algorithms for their particular need or task by e. g. emphasizing the pathways of interest by making them straight.
4. The algorithms do not sufficiently support interactive network exploration. Usually with these algorithms small modifications in the network structure and re-layout of the network results in very different pictures.

However, such sudden and large changes destroy the user's mental map (i. e. the user's understanding of the network based on the previous view) and therefore hinder interactive understanding of the network. Here I present a new algorithm for layout of biological networks that overcomes these limitations. It is based on a powerful new graph drawing technique, constrained graph layout [22]. Like force-directed layout [12,13] constrained graph layout works by minimizing an objective function that measures the quality of the layout. However it extends force-directed layout by allowing minimization of the objective to be done subject to placement constraints on the objects in the network. This is achieved by using mathematically rigorous optimization techniques based on gradient projection [23]. Efficient implementation is made possible by restricting the placement constraints to be separation constraints of the form \( u + g \leq (\neq) v \), enforcing a minimum (or precise) gap \( g \) between the positions \( u \) and \( v \) of pairs of objects in either the x or y dimensions of the drawing.

www.iosrjournals.org 45 | Page
The presented approach provides a generic, universal algorithm for layout of biological networks:
1. It greatly simplifies the implementation of layout methods for life sciences, systems and synthetic biology tools, which have previously had to utilize very different layout algorithms for different types of biological networks (or different layout requirements).
2. It allows the use of different layout styles for different parts of one large network.
3. It allows the user to customize the layout by adding separation constraints.
4. It lends itself to mental-map-preserving dynamic layout in interactive systems, thereby supporting interactive exploration of large and complex networks.

**Introduction**

A network is defined as a set of elements called vertices or nodes having connections among them called edges. Internet, the world wide web, Social networks (connection among individuals), networks of business relations, neural networks, food webs are examples of network.

The study of networks in the form of mathematical graph theory, is one of the fundamental pillars of discrete mathematics. Euler’s celebrated 1735 solution of the Konisberg bridge problem is cited as the first true proof in theory of networks.

**Types of Networks**

There are many ways of categorizing the network. Such as a network can have more than one type of different vertex or more than one different type of edge. If we take the example of social network of people, vertices may be men or women. People of different nationalities, locations, ages, incomeset. Edges may represent friendship, animosity or geographical proximity.

They can carry weights, representing how well two people know each other. They can also be directed, pointing in only one direction. Graphs composed of directed edges are themselves called directed graphs or sometimes digraphs.

A graph representing telephone calls or email messages between individuals would be directed. Since each message goes in only one direction. Directed graphs can be cyclic or acyclic. One can also have hyperedges - edges that join more than two vertices together. Graphs containing such edges are called hypergraphs. For example in social network - n individuals connect to each other by virtue of belonging to the same family can be represented by n-edge joining them.

**Glossary of terms**

Vertices- The fundamental unit of a network also called a site (physics), a node (Computer Science), or an actor (Sociology).

Edge-The line connecting two vertices. Also called a bond (physics), a link (Computer Science) or a tie (Sociology).

Directed/Undirected-An edge is directed if it runs in only one direction and undirected if it runs in both directions.

Degree-The number of edges connected to a vertex. A directed graph has both an in-degree and an out-degree for each vertex, which are the numbers of incoming and outgoing edges.

Component-The component to which a vertex belongs is that set of vertices which can be reached from.

In a directed graph a vertex has both an in-component (set of vertices from which the vertex can be reached) and out-component (set of vertices which can be reached from it). Geodesic path - Shortest path through the network from one vertex to another. Diameter - (number of edges) of the longest geodesic path between any two vertices.

Social Network - A Social network is a social structure made up of a set of social actors (such as individuals or organizations) and a set of dyadic ties between these actors. The social network perspective provides a set of methods for analyzing the structure of whole social entities as well as a variety of theories explaining the patterns observed in these structures. The study of these structures uses social network analysis to identify local and global patterns, locate influential entities and examine network dynamics. A social network is a set of people or groups of people with some pattern of contacts or interactions between them. The patterns of friendships between individuals, business relationships between companies, and intermarriages between families.

**Information networks**

Information networks sometimes called as knowledge networks. The classic example of an information network is the network of citations between academic papers. These citations form a network in which the vertices are articles and a directed edge from article A to article B indicates that A cites B. Citation networks are acyclic because papers can only cite other papers that have already been written, not those that have to be written.
Technological Networks

The man-made networks designed typically for distribution of resources such as electricity or Information for example electric power grid or Internet or telephone network.

Biological Networks

Biological processes are often represented in the form of networks such as protein-proteininteraction networks and metabolic pathways.

II. Basic Network features

The Small World Effect

A node’s degree or connectivity, giving the number of links k the node has, is the most elementary network measure. For example in following fig. nodes and J have exactly three links (k=3). The overall graph is characterized by average degree <k>, which has the value <k>=2.6.

As in most networks, there are multiple paths between any two nodes i and j. A useful distance measure is the length of the shortest path lij. The mean path length defined as

\[
\frac{1}{N(N-1)} \sum_{i \neq j} l_{ij}
\]

number of steps is often referred to display the ‘small world’ property, first illustrated on social networks, indicating that two randomly chosen individuals can be connected by only six intermediate acquaintances.

Transitivity or Clustering

In many networks it is found that if it is found that if vertex A is connected to vertex and vertex B is connected to vertex C, then there is a probability that vertex A will also be connected to vertex C. In terms of network topology, transitivity means the presence of a number of triangles in the network sets of three vertices each of which is connected to each of which is connected to each of the others. It can be quantified by defining a clustering coefficient C thus:

\[
C = \frac{3 \times \text{number of triangles in the network}}{\text{number of connected triples of vertices}}
\]

Where a “Connected triple” means a single vertex with edges running to an unordered pair of others.

In effect, C measures the fraction of triples that have their third edge filled in to complete the triangle. The factor of three in the numerator accounts for the fact each triangle contributes to three triples and ensures that C lies in the range 0 ≤ C ≤ 1. In simple terms, C is the mean probability that two vertices that are network neighbors of the same other vertex will themselves be neighbors. It can also be written in the form.

\[
C = \frac{6 \times \text{number of triangles in the network}}{\text{number of paths of length two}}
\]

Where a path of length two refers to a directed path starting from a specified vertex.

Degree distributions

The degree of a vertex in a network is the number of edges incident on (i.e. connected) to that vertex. We define Pk to be the fraction of vertices in the network that have degree k. Equivalently, Pk is the probability that a vertex chosen uniformly at random has degree k. If a network is directed, meaning that edges point in one direction from one node to another node, then nodes have two different degrees, the in-degree which is the number of incoming edges, and the out-degree which is the number of outgoing edges.

The degree distribution p(k) of a network is then defined to be the fraction of nodes in the network with degree k. Thus if there are n nodes in total in a network and nk of them have degree k, we have p(k)=n/k/n.

The degree distribution is very important in studying both real networks, such as the Internet and social networks, and theoretical networks. The simplest network model, for example, the (Bernoulli) random graph, in which each of n nodes is connected (or not) with independent probability p (or 1-p), has a binomial distribution of degree k (or Poisson in the limit of large n). Most networks in the real world, however, have degree distributions very different from this. Most are highly right skewed, meaning that a large majority of nodes have low degree but a small number, known as “hubs” have high degree.

Biological Networks

Biological processes are often represented in the form of networks such as protein-protein interaction networks and metabolic pathways. The study of biological networks, their modeling, analysis, and visualization are important tasks in life science today. An understanding of these networks is essential to make biological sense of much of the complex data that is now being generated. This increasing importance of biological networks is also evidenced by the rapid increase in publications about network-related topics and the growing
number of research groups dealing with this area. Most biological networks are still far from being complete and they are usually difficult to interpret due to the complexity of the relationships and the peculiarities of the data. Network visualization is a fundamental method that helps scientists in understanding biological networks and in uncovering important properties of the underlying biochemical processes. This chapter therefore deals with major biological networks, their visualization requirements and useful layout methods. We start with some basic biology and important biological networks.

Molecular Biological Foundations

A cell consists of many different (bio-) chemical compounds. A crucial macromolecule in organisms is DNA (deoxyribonucleic acid), which is the carrier of genetic information. But DNA itself is not able to provide the structure of a cell, to act as a catalyst for chemical reactions or to sense changes in the cell’s environment. Such functions are carried out by proteins, large molecules which are built according to information stored in DNA sequences.

The central dogma of molecular biology deals with the information transfer from DNA to proteins. It states that proteins do not code for the production of other proteins, DNA or RNA (ribonucleic acid), i.e., that information cannot be transferred from one protein to another protein directly or from a protein back to nucleic acid. Instead, the standard pathway of information flow is from DNA to RNA to protein. Genes represented by DNA sequences are transcribed into RNA sequences which are then translated into proteins, see Figure 20.1. These proteins have different types such as structural components (which give cells their shape and help them move), transport proteins (which carry substances such as oxygen), enzymes (which catalyze most chemical processes in cells and help change metabolites into each other) and regulatory proteins (which regulate the expression of other genes). Crick summarized the standard pathway of information flow as “DNA makes RNA, RNA makes protein and proteins make us” [Kel00].

Signal Transduction and Gene Regulatory Networks

A key issue in biology is the response of a cell to internal and external stimuli and the subsequent regulation of its genetic activity. Signal transduction and gene regulatory pathways and networks describe processes to coordinate the cell’s response to such stimuli. Here we consider both networks together as the underlying mechanisms have many similarities, the networks share some common elements and both often result in the regulation of gene expression. Consequently, similar visualization approaches are used for signal transduction and gene regulatory pathways and networks.

Definition

Signal transduction is a communication process within a cell to coordinate its responses to an environmental change. The stimulus comes from the cell’s environment, e.g., molecules such as hormones. The response is a reaction of the cell, e.g., the activation of a gene or the production of energy. A signal transduction pathway is a directed network of chemical reactions in a cell from a stimulus (an external molecule which binds to a receptor on the cell membrane) to the response (e.g., the activation of a gene). Here we focus on signal transduction pathways that aim at transcription factors and thus alter the expression of genes in a cell. The signal transduction network of a cell is the complete network of all signal transduction pathways. A signaling cascade is a process where signal transduction involves an increasing number of molecules in the steps from the stimulus to the response.

Gene regulation is a general term for cellular control of the synthesis of proteins at the transcription step. Gene regulation can also be seen as the response of a cell to an internal stimulus. Often one gene is regulated by another gene via the corresponding protein (called transcription factor), thus gene regulation is coordinated in a gene regulatory network. This network directs the level of expression for each gene in the cell by controlling whether and how often that gene will be transcribed into RNA. Similar to signaling cascades in...
signal transduction networks a gene can activate more genes in turn and an initial stimulus can trigger the expression of large sets of genes.

As mentioned above we study signal transduction and gene regulation together. Figure 20.1 sketches both processes with signal transduction going from an external signal via several steps to the activation of a gene as one possible response and gene regulation going from a gene via a protein to another gene. Events of signal transduction and gene regulatory processes occur in different parts of a cell (cellular compartments). To represent compartments these networks can be modeled as clustered graphs. A clustered graph $C = (G, T)$ consists of a directed graph $G = (V, E)$ and a rooted tree $T$, such that the leaves of $T$ are exactly the nodes of $G$. The nodes $v \in V$ of the graph are chemical and biochemical compounds (ranging from ions, to small molecules, macromolecules and genes) and the edges $e \in E$ are biochemical events (e.g., binding, transportation and reaction). The occurrence of signal transduction and gene regulatory events in different cellular compartments can be modeled by the tree $T$. Each node $t \in T$ represents a cluster of nodes of $G$ consisting of the leaves of the sub tree rooted at $t$. The modeling of such networks based on clustered graphs can be used for cluster-preserving layout algorithms [EH00]. However, as it is only partly known in which compartment an event occurs, signal transduction and gene regulatory processes are usually modeled by graphs. The pathways and networks can be derived from databases such as KEGG [KGKN02, KGH+06] and

Gene Regulatory Network

The entities are subdivided into 4 classes: 1) Protein or protein complex; 2) Gene; 3) RNA; 4) Nonproteinaceous Substance. Instances of each class are described in a separate table in the GeneNet database. The components of a gene network are scattered throughout cellular compartments, cells. Two types of relationships between the entities are considered: Reaction, that is, formation of a new entity or acquisition of a new property by the entity, and Regulatory event, that is, the effect of an entity onto a certain reaction.

Protein Protein Interaction

While traditional biochemical experiments had generated a small set of data for individual protein–protein interactions [34], the last three years have seen a rapid expansion of protein interaction data due to the recent development of high-throughput interaction detection methods such as yeast two-hybrid (Ito et al., 2000) and mass spectrometry techniques. The interaction data is available either in text files or in databases. However, due to the volume of data, a graphical representation of protein interactions has proven to be much easier to understand than a long list of interacting proteins. Furthermore, a network of protein interactions provides us with a clear notion of protein function by showing a context within which function can be interpreted. Protein–protein interactions are typically visualized as an undirected graph $G = (V, E)$, where $x, y \in V$ represent *To whom correspondence should be addressed.* Proteins and $(x, y) \in E$ represents an interaction between proteins $x$ and $y$. Visualization of a graph is straightforward when dealing with a small number of nodes and edges. In practice, protein–protein interaction networks often consist of thousands of nodes or more, which severely limit the usefulness of many graph drawing tools either because they produce cluttered drawings with many edge crossings or static drawings that are not easy to modify, they are too slow for interactive analysis with large data sets, or because they require input data to be in specific format rather than taking the data directly from protein–protein interaction databases. The ultimate usefulness of a protein interaction network depends on the readability of the network, and therefore, a protein interaction network should focus on conveying the interaction information quickly and clearly.

Force-directed layout algorithms have been the most popular methods for visualizing an undirected graph, which produce an optimal layout based on a force model. A simple implementation of a force-directed algorithm encounters read difficulties when drawing graphs of more than a few hundred nodes. These difficulties originate from two sources. First, layout adjustment involves computation of force between every pair of nodes at each step of the optimization process. Second, for large graphs the optimization process needs too many iterations for transforming the initial random layout into an optimal layout. Previously we developed a force-directed layout...
program called InterViewer (Juet al., 2003). In this paper, I present a new program that efficiently produces a protein interaction network of good quality without computing force between every pair of nodes. This improves InterViewer in many ways: (1) while InterViewer produces a drawing by computing force between every pair of nodes in each iteration of the optimization process, this produces a more pleasant drawing without computing force between every pair of nodes, (2) this is faster than InterViewer, (3) this provides several abstraction operations to reduce complex networks into simpler ones, and (4) multiple protein interaction networks can be compared for common proteins and their interactions shared by all or part of the networks.

**Algorithm for protein interaction networks**

Mid nodes (v5–v11) on three paths between a pair of enclosing cut vertices (c1 and c2). Since the multiple paths have different lengths, mid nodes on the paths are grouped into two groups: mid node group 1 = {v5, v6, v8, v9}, mid node group 2 = {v7, v10, v11}.

### III. Definitions

The degree of a node v is the number of its edges and is denoted by deg(v). A cutvertex (also called an articulation point) in a graph G is a node whose removal disconnects G. A path in a graph G is a sequence (v1, v2, . . . , vn) of distinct nodes of G, such that (vi , vi+1) ∈ E for 1 ≤ i ≤ n − 1. A graph G_ = (V_, E_), such that V_ ⊆ V and E_ ⊆ E ∩ (V_ × V_), is a subgraph of graph G = (V , E).

When multiple paths exist between a pair of cut vertices, we call the nodes on the paths mid nodes. In Figure there are mid nodes (shown in yellow) on three paths between a pair of enclosing cut vertices (shown in blue). If the multiple paths between a pair of cut vertices have different lengths, mid nodes on the paths of the same length are grouped together.

What we call pivot nodes are the key nodes in the layout of a graph. In order to produce a layout of high quality efficiently, we select pivot nodes that are almost uniformly distributed in each connected component (see Fig. 2 for examples). The number of pivot nodes and distance between them are determined based on the number of nodes and edges, and the diameter of a connected component (a diameter of a connected component is the maximum distance between two nodes in the component). In general, more pivot nodes are selected for a connected component with a large diameter compared to the number of nodes than for a connected component with a small diameter compared to the number of nodes. For a small connected component that has 100 nodes or fewer, we select more pivot nodes so that the distance between them may be 3 or less. However, each connected component can have at most 100 pivot nodes in any case for the efficiency of the algorithm.

A detailed method for selecting pivot nodes and for computing the distance between them is described in Algorithms 2 and 3 later.

### IV. The Algorithm

A common problem with many force-directed layout algorithms is that they become very slow when dealing with large graphs because layout adjustment at each step typically involves computation of force between every pair of nodes. Since a protein interaction network tends to be a disconnected graph with several connected components, we first compute a layout of connected components and then compute a layout of nodes within a connected component. Our experience is that this approach produces much better drawings in a shorter time than computing a layout of all nodes from the beginning. Our algorithm uses a multilevel technique to draw a graph. It is composed of two steps at the top level: grouping and layout.

In the grouping step, the algorithm first groups nodes of a disconnected graph into connected components, and finds mid nodes and pivot nodes in each connected component. In the layout step, the coarsest graph is an initial layout of connected components based on their pivot nodes only. The layout of each connected
component is then refined locally within the connected component based on its mid nodes and neighbors of each node. Each step of the algorithm can be summarized as follows.

1. **Grouping**
   (a) Identify all connected components of an entire network.
   (b) For each connected component, determine its mid nodes and pivot nodes.
   (c) Compute the distance of every node from the pivot nodes of the connected component to which the node belongs.

2. **Layout**
   (a) Find a layout of connected components of an entire network (layout between connected components).
   (b) For each connected component find a layout of nodes with respect to the pivot nodes of the connected component (global layout within a connected component).
   (c) Refine the layout of each connected component by relocating the mid nodes adjacent to cut vertices with respect to the cut vertices and the cut vertices’ direct neighbors (local layout of mid nodes within connected component).
   (d) Refine the layout of each connected component by relocating all nodes with respect to their neighbors within distance 2 (local layout of all nodes within a connected component).

Step 1(a) is straightforward, and Algorithm 1 describes step 1(b). In Algorithm 1, a group represents a connected component. Since step 1(a) and Algorithm 1 are performed on nodes with at least one edge, nodes with no edge are positioned after the connected components of size ≥ 2 are positioned in step 2(a). For a graph with |V| = n nodes, the time complexity of step 1(a) is O(n), and the time complexity of Algorithm 1 is O(|PvN| · n).

**Algorithm 1**

```plaintext
Algorithm 1 Distance(v, w)
1: DLst.Add(v, 0)  \{Add v and its distance (= 0) from v to DLst\}
2: DLst.First \{Get the first node of DLst\}
3: repeat
4: DLst.GetCurrent(v_, currentDist)  \{Get the current node v_ and its distance from v\}
5: for all neighbor u of v_ do
6: if u ∈ DLst then
7: if w = u then
8: return currentDist + 1 \{distance between v and u\}
9: end if
10: DLst.Add(u, currentDist + 1)  \{Add u and its distance from v to DLst\}
11: end if
12: end for
13: DLst.Next \{Get the next node of DLst\}
14: until DLst.Eof \{until no more nodes exist in DLst\}
```

Selecting pivot nodes from each connected component in step 1(c) is done by Algorithms 2 and 3. When selecting pivot nodes, distances of the pivot nodes from all other nodes are also computed. Algorithms 2 and 3 take O(n) time for a single pivot node, and therefore, the total time complexity for selecting all pivot nodes is O(|PvN| · n). Algorithm 3 examines whether the current node v is already a pivot node; if not, it determines the possibility of including the node to the pivot node set PvN depending on the distance from existing pivot nodes, the structure of the connected component (i.e., diameter, number of nodes and edges of the connected component).

The current node v can be selected as a pivot node if

**Algorithm 2**

```plaintext
Algorithm 2 SelectPivotNodes
1: MaxDist ← 1
2: PvN.Add(V[0], DistTable.Create(V[0], 0))  \{first node in a group\}
3: PvN.First \{Get the first node of PvN\}
4: repeat
5: DLst.Clear \{Initialize DLst as an empty list\}
6: DLst.Add(PvN.CurrentPivotNode, 0) \{Add the current pivot node and its distance\}
7: DLst.First \{Get the first node of DLst\}
8: repeat \{distance from pivot nodes\}
```

![Fig. 2](image-url) (a) Pivot nodes (shown in green) selected from a mesh. (b) Pivot nodes (shown in green) selected from a protein interaction network.
9: ChkDistance(DLast, PvN.CurrentDistTable, MaxDist)
10: DLast.Next {Get the next node of DLast}
11: until DLast.Eof {until no more nodes exist in DLast}
12: PvN.Next {Get the next node of PvN}
13: until PvN.Eof {until no more nodes exist in PvN}

it satisfies the following rules (function ChkPvN(v)in step 16 of Algorithm 3).
1. In a connected component with < 40 nodes, the distance of v from all existing pivot nodes should be at least 2.
2. In a connected component with ≥ 40 and < 100 nodes, the distance of v from all existing pivot nodes should be at least 3.
3. In a connected component with ≥ 100 nodes, 
   (a) if the diameter (d) of the connected component is < 7, degree(v) should be ≥ 3.
   (b) if 7 ≤ d < 15, degree(v) should be ≥ 4.
   (c) if 15 ≤ d < 20, degree(v) should be ≥ 5.

Algorithm for protein interaction networks
Algorithm 3 ChkDistance(DLast, DistTable, MaxDist)
1: DLast.GetCurrent(v, dist) {Get a node v and its distance from a pivot node}
2: if (dist > MaxDist) then
3: MaxDist ← dist {Update the maximum distance}
4: end if
5: bAddPvN ← true {potential pivot node}
6: for all neighbor w of v do
7: if w ∈ DLast then {distance of w from a pivot node has not been determined.}
8: bAddPvN ← false {w cannot be a pivot node}
9: DLast.Add(w, dist + 1) {Add w and its distance from v to DLast}
10: DistTable(w) ← dist + 1 {Store the distance of w from v in DistTable}
11: end if
12: end for
13: if MaxDist/3 = dist then {The node is at a distance of one third of the maximum distance}
14: bAddPvN ← true {potential pivot node}
15: end if
16: if bAddPvN and ChkPvN(v) then
17: PvN.Add(v, DistTable.Create(v, 0))
18: end if
(d) else, let R be the ratio of the diameter of the connected component to the number of nodes of the connected component.
(i) if R < 0.01, the distance of v from all existing pivot nodes should be at least 40.
(ii) if 0.01 ≤ R < 0.02, the distance of v from all existing pivot nodes should be at least 17. If the total number of nodes > 1000, adjust the distance to 30.
(iii) if 0.02 ≤ R < 0.035, the distance of v from all existing pivot nodes should be at least 13. If the total number of nodes > 1000, adjust the distance to 20.
(iv) if 0.035 ≤ R < 0.07, the distance of v from all existing pivot nodes should be at least 10.
(v) if R ≥ 0.07, the distance of v from all existing pivot nodes should be at least 5.

Algorithm 4 provides a concise description of all layout styles of step 2, including both global layout and local layout. The position of v is always determined with respect to a referenceset V_, which is a subset of V. In step 2(a), the referenceset V_ is a set of pivot nodes of other connected components, to which v does not belong. The maximum diameter of all connected components is used as the value of Distance(u, v) in step 4 of Algorithm 4, and therefore is constant for all nodes. In step 2(b), the reference set V_ is a set of pivot nodes.

Algorithm 4 Layout(v, V_)
1: D ← 0 {Initialize the position displacement D to 0}
2: for all u ∈ V_ do {V_: subset of V}
3: _ ← pos[u] − pos[v] {pos[u]: position of node u}
4: D ← D + (_(1 − Distance(u, v)/_))
Metabolic Networks

Metabolic reactions are fundamental to life processes, e.g., for the production of energy and the synthesis of substances. A huge number of reactions occur at any time in living cells and the product of one reaction is usually used by another reaction, thus metabolic reactions are strongly interconnected and form metabolic pathways and networks.

A metabolic reaction \( R \) is a transformation of chemical substances or metabolites (reactants) into other substances (products) usually catalyzed by enzymes. In general, metabolic reactions are reversible, that is, they occur in both directions. Such reactions are characterized by a steady state, i.e., if occurring isolated they reach a state where the amount of change in both directions is equal. A cell is in a constant exchange of substances with its environment. Furthermore, many reactions are regulated, i.e., they are suppressed or enhanced by other factors (allosteric control). This shifts the steady state and together with the steady supply of substances from outside and their final use, e.g., by exporting them from the cell, one can consider a main direction of a reaction. This is also expressed by the differentiation of substances into reactants and products. As already seen, metabolic reactions interact with each other, i.e., the product of one reaction is usually a reactant of another reaction. A metabolic path \( P = (R_1, \ldots, R_n) \) is a sequence of metabolic reactions where for all \( 1 \leq i < n \) at least one product of reaction \( R_i \) is a reactant of reaction \( R_{i+1} \).

The metabolic network or metabolism of a particular cell or an organism is the complete network of metabolic reactions of this cell or organism. A metabolic pathway is a connected sub-network of the metabolic network either representing specific processes or defined by functional boundaries, e.g., the network between an initial and a final substance as shown in Figure 20.5.

From a formal point of view, a metabolic pathway is a hyper-graph. The nodes represent the substances and the hyper-edges represent the reactions. A hyper-edge connects all substances of a reaction, is directed from...
reactants to products and is labeled with the enzymes that catalyze the reaction. Hyper-graphs can be represented by bipartite graphs.

Additionally to the nodes representing substances, the reactions are nodes (either labeled with the enzymes or with further nodes for enzymes) and edges are binary relations connecting the substances of a reaction with the corresponding reaction node. This is a common modeling of metabolic pathways, e.g., for their simulation using Petri-nets [HT98, RML93].

For the analysis and visualization of metabolic pathways, substances are often divided into two types [MZ03]: main substances and co-substances. Co-substances are usually small and short metabolites, e.g., ATP, ADP, H2O, NH3 and NADH. These substances normally transfer electrons or functional groups such as phosphate and amino groups [NIS90]. Main substances are all other metabolites. However, this is not a global property but is given according to the reaction [MZ03], and a small metabolite such as ATP may be considered main substance in a particular reaction. For visualization purposes this distinction is important as main substances and co-substances are often differently visually represented.

Here a metabolic pathway is modeled as directed bipartite graph G = (VS, VR, E) with nodes u1, . . . , un reaction nodes (either labeled with the enzymes catalyzing the reaction) and directed edges (u1, v), . . . , (un, v) ∈ E representing the transformation of substances u1, . . . , un to substances v1, . . . , vn by the reaction v. A reversible reaction does not contain backward edges as in some models for simulation purposes, instead this property of a reaction is represented by an attribute. Another attribute is used to mark main and co-substances.

Types of Metabolic Networks

• Simplified metabolic network: A network which contains reactions, enzymes and main substances, but no co-substances.

• Metabolite network and simplified metabolite network: A network which consists only of substances (metabolites), in the simplified case only of main substances.

• Enzyme network: A network which consists only of the enzymes catalyzing there actions.

References

[21]. S. F. Carrizo. A survey of phylogenetic researchers: A network which consists only of the enzymes catalyzing there actions.


[34]. A fast layout algorithm for protein interaction. Networks Kyungsook Han and Byong-Hyon Ju School of Computer Science & Engineering,Inha University,Inchon 402-751,Korea Received on May 1,2003;revised and accepted on July 15,2003