



HAL
open science

Pathway Preserving Representation of Metabolic Networks

Antoine Lambert, Jonathan Dubois, Romain Bourqui

► **To cite this version:**

Antoine Lambert, Jonathan Dubois, Romain Bourqui. Pathway Preserving Representation of Metabolic Networks. Computer Graphics Forum, 2011, Special Issue Bergen, Norway 1 - 3 June 2011, 30 (3), pp.1021 - 1030. inria-00579267

HAL Id: inria-00579267

<https://inria.hal.science/inria-00579267>

Submitted on 28 Feb 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Pathway Preserving Representation of Metabolic Networks

A. Lambert¹ and J. Dubois¹ and R. Bourqui¹

¹CNRS UMR5800 LaBRI, Université Bordeaux 1 and INRIA Bordeaux - Sud Ouest, France

Abstract

Improvements in biological data acquisition and genomes sequencing now allow to reconstruct entire metabolic networks of many living organisms. The size and complexity of these networks prohibit manual drawing and thereby urge the need of dedicated visualization techniques. An efficient representation of such a network should preserve the topological information of metabolic pathways while respecting biological drawing conventions. These constraints complicate the automatic generation of such visualization as it raises graph drawing issues. In this paper we propose a method to lay out the entire metabolic network while preserving the pathway information as much as possible. That method is flexible as it enables the user to define whether or not node duplication should be performed, to preserve or not the network topology. Our technique combines partitioning, node placement and edge bundling to provide a pseudo-orthogonal visualization of the metabolic network. To ease pathway information retrieval, we also provide complementary interaction tools that emphasize relevant pathways in the entire metabolic context.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation

1. Introduction

The metabolism is the set of biochemical reactions that occur in a living system. Each reaction transforms a set of molecules (or metabolites) called substrates into other molecules called products. In biological systems, the reactions are activated (catalyzed) by enzymes, composed by proteins that are coded by genes [LCTS08].

When interested in metabolisms, one can consider different scales that vary according to the data and the biological questions. For instance, toxicologists often follow the degradation of a given molecule; in that case they only focus on a very small number of reactions. At a larger scale, biologists studying glycolysis will focus on a particular set of biochemical reactions allowing the organism to perform that specific biological function. Such a set of reactions is called a *metabolic pathway*. Most of the work on metabolism visualization has been done at this level of detail [BR01, Sch03, BDS03, WK05, GZ05, DEG*05, KGF*10, MWS*10]. Due to the hierarchical organization of metabolic pathways, a common approach consists in using hierarchical (or layered) drawing algorithms (e.g. [STT81, Aub03]). For instance, the algorithms of Sirava *et al.* [SSE*02], Schreiber [Sch03] but also Brandes *et al.* [BDS03] lay out pathways in a lay-

ered manner. Nevertheless, these approaches do not fit biologists expectations as they do not respect one of the most commonly used biological convention: circular representation of cycles of reactions. To overcome that issue, some other algorithms [BR01, WK05, GZ05] first detect cycles in the pathway, draw them using a circular layout algorithm, collapse them into metanodes, repeat that process until no more cycle is detected and finally draw the resulting graph using either force-directed [BR01, WK05] or dedicated hierarchical [GZ05] drawing algorithms. Recently, Meyer *et al.* [MWS*10] proposed a new visualization method called linearized metabolic pathways view that does not represent pathways as graphs but rather as segments corresponding to sequences of elements computed during a linearization process.

These methods simultaneously represent several metabolic pathways (up to five according to [BR01]) but cannot be used to draw the largest scale of metabolism studies that is the *metabolic network*. The metabolic network of an organism is the result of the integration of all its pathways into a single network [KOMK*05, KCVGC*05]. The issue of analyzing biological processes spanning several metabolic pathways appears in many contexts. In particular,

it is very likely to appear every time the experiments are not pathway-focused. This is generally the case for expression data or metabolomics data. Therefore, pathway visualization is not suitable for such tasks but neither is network visualization without any pathway information. Indeed, to be useful for mapping experiments, it is necessary to represent the entire network structure while keeping the contextual information provided by its division into metabolic pathways.

First approaches to represent the entire metabolic network consist in using classical drawing algorithms such as in *SBML Viewer* [SBM] or *Cytoscape* [SMO*03]. In these tools, metabolic networks are usually laid out using force-directed algorithms. However, the layouts computed by these algorithms do not respect the biological drawing conventions, therefore such representations are not fully satisfying [SND05] from a biologist's point of view. This is particularly true when using force-directed algorithms as visually dense regions of the drawing do not correspond to single pathways but rather to sets of pathways. Moreover metabolic pathways are not necessarily connected subnetworks so that the task of identifying a single pathway is made difficult.

Only a few tools that respect biological drawing conventions [JTG*05,PK06,BLC*07,RHR*10] exist. In both *Reactome* [JTG*05] and *Pathway Tools cellular overview diagram* [PK06], nodes shared by several pathways are duplicated and each pathway is laid out independently. Finally all the pathway drawings are embedded in a single view according to their common functions. In [RHR*10], Rohrschneider *et al.* also use node duplication. In that approach, nodes are embedded on a regular orthogonal grid and edges are routed on that grid using a cost function minimization technique. To reduce clutter of the visualization, the authors allow routing of edges on the same segments resulting in an edge bundled visualization. Finally each pathway is collapsed into a single metanode and a compound graph visualization is provided to the user (who can expand/collapse these pathways when required). However, none of these tools provides a visualization that reflects the metabolic network topology as shared nodes are duplicated. To the best of our knowledge, only one tool exists, it is called *MetaViz* [BLC*07] and keeps unchanged the network topology while emphasizing its decomposition into pathways. This is achieved first by turning the overlapping decomposition of pathways into a partition of nodes, then by using dedicated drawing algorithms to draw each computed cluster and next by embedding these clusters using a planar drawing algorithm. Finally a multi-scale compound graph visualization is provided to the user. Such a visualization reduces the clutter but is not fully satisfying as non-expert users (and in particular biologists) are not used to navigate that kind of abstracted view.

In this paper we propose a method to lay out the entire metabolic network while preserving the pathway information as much as possible. That method is flexible as

it enables the user to define whether or not node duplication should be performed, to preserve or not the network topology. Our technique makes a trade-off between the approach of Bourqui *et al.* [BLC*07] and the approach of Rohrschneider *et al.* [RHR*10]. It combines node partitioning (similar to [BLC*07]), node placement and edge bundling (like [RHR*10]) to provide a pseudo-orthogonal visualization of the metabolic network. To ease pathway information retrieval, we also provide complementary interaction tools that emphasize relevant pathways in the entire metabolic context.

The remainder of this paper is structured as follows. In section 2, we present the main challenges in metabolic network visualization. We next give the details of our graph drawing algorithm in section 3. Then, we describe two interactions supporting pathway information retrieval in section 4. Finally, we draw a conclusion and give directions for future work in section 5.

2. Metabolic network drawing : Challenges

Biological drawing conventions have been established to ease information retrieval in the representation of biological processes. Such conventions can be observed in hand-drawn representations [Mic93,Nic97,Mic98]. One can first notice in these drawings that only few metabolites and reactions have more than one occurrence. Biologists also use circular (respectively straight-line) representations of cycles (respectively cascades) of reactions. Next, as mentioned in [RHR*10], edges are drawn in a pseudo-orthogonal manner. Finally, one can notice in hand-drawn representations that information related to the decomposition of the network into pathways should be preserved as each pathway is laid out in a close region of the drawing. One can also notice that edges are drawn as short as possible and that the number of edge crossings is minimized.

The main challenge when drawing a metabolic network is to respect these conventions in an automatic drawing algorithm. To reach that aim, we translated some of these requirements into 5 constraints:

- **Closeness constraint:** Pathway information has to be preserved as much as possible
- **Duplication constraint:** Avoid node duplications in order to provide a visualization that respects network connectivity
- **Topological constraint:** Particular representations of cascades and cycles of reactions
- **Edge drawing constraint:** pseudo-orthogonal drawing of edges
- **Clutter constraint:** Minimization of edge crossings or at least an uncluttered representation of the network

Each of these constraints raises either well-known computational or graph drawing problems. To respect the *Topological constraint*, we need to search for cycles in the network

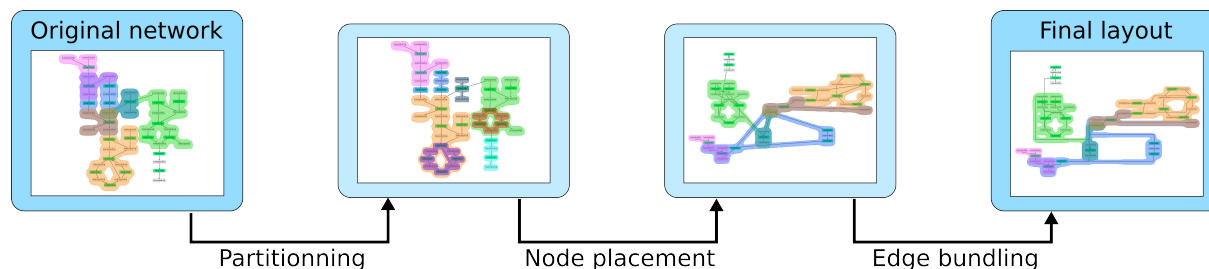


Figure 1: Drawing pipeline of our method. First step turns an overlapping decomposition into a partition, then nodes are laid out and finally edges linking different clusters are routed using an edge bundling technique.

which is known to be an NP-complete problem [Kar72]. Minimizing the number of edge crossing is also known to be NP-complete [GJ83]. Concerning the *Closeness constraint*, the main problem comes from the fact that pathways often share metabolites and/or reactions. Suppose for instance that our network contains 5 pathways, each sharing elements with the 4 others. Then, there is no 2D drawing of that network with no unexpected pathways overlap. Therefore, respecting that constraint, while avoiding node duplication, is not straightforward.

Our method draws the entire network while trying to respect these constraints simultaneously. In many cases, no representation can fulfill all those requirements. In particular, a trade-off has to be found between the *Closeness constraint* and the *Duplication constraint*. Depending on his task, the user can decide in our method whether or not node duplication should be performed. It guarantees that the *Closeness constraint*, or on the contrary that the *Duplication constraint*, will be respected.

3. Network drawing framework

In the following, metabolic networks are modeled as bipartite graphs $G = (V, E)$ where $V = R \cup M$ and R is the set of reactions and M the set of metabolites, and $E \subseteq R \times M$ (for a detailed discussion about metabolic network models, refer to [vWGW02]).

3.1. Method pipeline

Figure 1 shows the main steps of our technique. First of all, the overlapping decomposition of the network (i.e., its decomposition into metabolic pathways) is turned into a partition of the elements. Then a compound graph is computed according to the created partition. During that step, our technique automatically determines pathways and subpathways that will respect the *Closeness constraint*.

During the node placement step, that compound graph is laid out in a bottom-up manner, i.e., deepest clusters are drawn first, then upper level clusters. It takes the area needed

to draw a cluster into account when drawing upper level clusters.

Finally, metanode expansion is performed and an edge bundling technique is applied to route inter-cluster edges in a pseudo-orthogonal manner. This also reduces the clutter in the representation, and therefore eases information retrieval.

3.2. Partitioning the network : unfolding the DAG

As mentioned above, a metabolic network is decomposed into overlapping metabolic pathways. A clustered (partitioned) graph is usually modeled by a pair (G, T_G) (see [EF96]) where G is a graph and T_G is the hierarchy tree representing the multilevel partition of the nodes. In case of overlapping clusters, we need to use a pair (G, D_G) where D_G is not a hierarchy tree but rather a hierarchy DAG [Bou08]. In Figure 2 (left), one can see 4 pathways sharing elements. For instance, pathways p_1 and p_2 share 4 elements (3 metabolites and 1 reaction). To model that inclusion property, the nodes representing pathways p_1 and p_2 in the hierarchy are both predecessors of these 4 shared elements (these multiple predecessors relationships are shown in red in the hierarchy).

However, drawing such a decomposed graph is not straightforward or may even be impossible without creating unexpected pathway overlaps (see [SAA09]). To solve that issue, we slightly modified the clustering algorithm of [BLC*07] that turns a hierarchy DAG into a hierarchy tree while preserving the pathway information as much as possible. That partitioning scheme is made of 3 main steps:

- **Computation of a maximal set of independent pathways** and clustering of *proper subgraphs* of the remaining pathways (i.e., the subgraph of a pathway induced by elements only belonging to that pathway). Then, elements that do not belong to any pathway are clustered together. In Figure 2 (middle-left), pathways p_1 and p_3 have been chosen as the independent set, proper subgraphs of p_2 and p_4 but also the remaining nodes have been clustered.
- **Connected components computation:** each cluster is decomposed into its connected components. In Figure 2

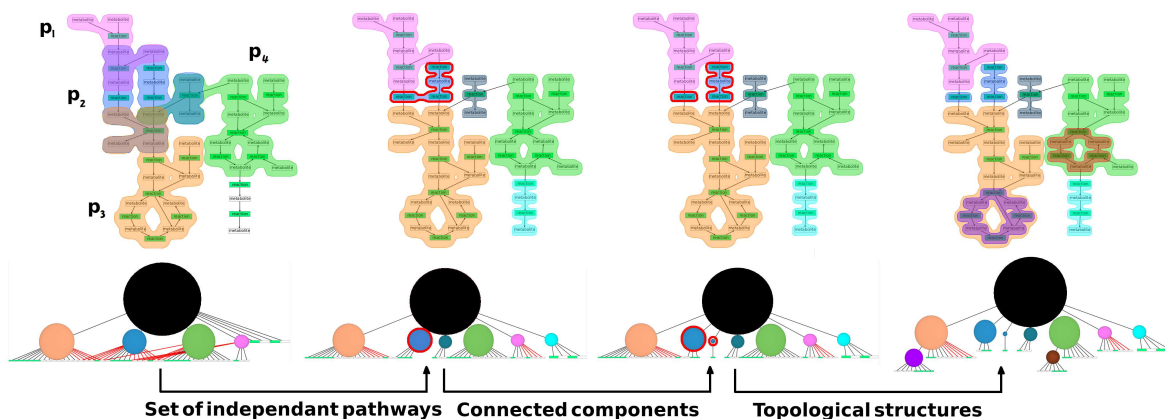


Figure 2: Illustration of the partitioning process. First, an independent set of pathways is computed and metabolites/reactions without pathways are clustered. Next, connected components are computed (see red surrounded nodes). Finally, topological structures are detected within each cluster.

(middle-right), a proper subgraph of p_2 (surrounded in red) has been decomposed into its two connected components.

- **Topological structures detection:** Cycles and cascades of reactions are clustered. In Figure 2 (right), two cycles have been detected.

One of the key features of this partitioning scheme is to enable the user to drive the computation of the set of independent pathways. By choosing pathways of interest, the user decides to enforce the Closeness constraint to be respected for these particular pathways.

3.3. Drawing the network

Our drawing method follows a bottom-up scheme, i.e., levels of the computed hierarchy are laid out from the deepest to the highest levels.

3.3.1. Building the multilevel compound graph

To achieve the drawing step of our technique, we first need to build a compound graph associated to the newly computed multilevel hierarchy of clusters. Given a partition of the original nodes, a compound graph is constructed as follows: for each cluster in the partition there is one *metanode* in the compound graph and two metanodes are linked by a *metaedge* if and only if there is at least one edge between the corresponding clusters.

In our approach, the compound graph corresponding to the highest level of the multilevel hierarchy is first built, then the process is repeated iteratively in the deeper levels (until all the levels have been abstracted) to result in a multilevel compound graph. Figure 3 shows the multilevel compound graph associated to the graph of Figure 2 and the computed hierarchy.

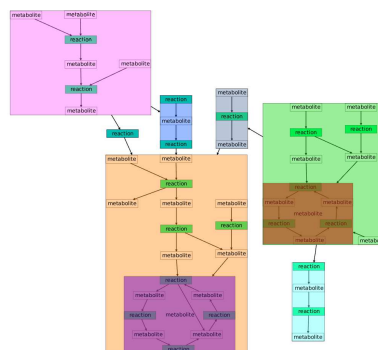


Figure 3: Multilevel compound graph associated with graph of Figure 2 and the computed multilevel hierarchy tree.

3.3.2. Drawing the deepest levels of the hierarchy

According to the partitioning scheme, clusters of the computed hierarchy are either (sub)pathways or particular topological structures. To lay out these clusters, our technique uses two drawing algorithms. First, layouts of cycles of reactions are computed by a circular drawing algorithm. Second, each other cluster is laid out by a hierarchical (or layered) drawing algorithm [Aub03]. Such an algorithm is particularly well-suited as it emphasizes cascades of reactions as well as the hierarchical organization of pathways. In the next step, our drawing process will position these clusters in the plan. To get a compact layout and therefore to increase information density, our process computes a horizontal and a vertical layout of these clusters (see section 3.3.3).

3.3.3. Drawing the top-level compound graph

Before drawing the top-level compound graph, metanode sizes are set to the area needed to lay out the corresponding clusters. Next, for each metaedge, we compute its anchors positions according to the positions of the source and target of the abstracted edges. The top-level compound graph is then laid out using a force-directed algorithm adapted from GEM [FLM94]. That algorithm can take nodes sizes but also anchored edges into account.

Another important feature of our force-directed algorithm consists of performing “rotations” of the metanodes to obtain a more compact layout. In that sense, it combines a force-directed algorithm and a Torque algorithm (see [ST04]). However, our algorithm only needs to check four configurations as to respect biological conventions clusters should be laid out vertically (top to bottom and bottom to top) or horizontally (left to right and right to left). Rather than rotating the metanodes, our approach modifies their sizes and edges anchors positions according to the computed vertical and horizontal layouts. At each step of the force-directed algorithm, each configuration is tested and the configuration that minimizes the average edge length is chosen.

By checking each of these four configurations, our algorithm lays out clusters vertically or horizontally, and therefore results in a drawing that is closer to hand-drawn representations. It also increases information density by providing more compact layouts.

3.4. Clutter Reduction : bundling inter-clusters edges

During the previous steps, clusters and the top-level compound graph have been laid out. To provide a classical node-link diagram visualization of the network, the next step consists of expanding the metanodes. However, our force-directed algorithm computes a straight-line drawing therefore inter-cluster edges, i.e., edges that are abstracted by metaedges in the top-level compound graph, are also drawn as single segments. It may lead to a cluttered representation that does not respect the *Edge drawing constraint* defined in section 2.

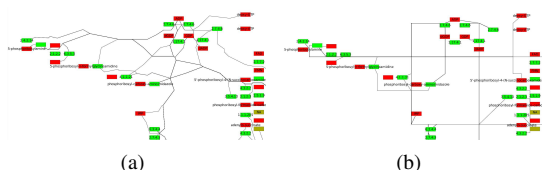


Figure 4: Zoomed view on yeast metabolic network (a) bundled with the original algorithm of Lambert et al. [LBA10], (b) only using a quadtree.

To both reduce the clutter and obtain a pseudo-orthogonal

drawing of the edges, our technique bundles inter-clusters edges. This is achieved by a dedicated version of [LBA10]. In [LBA10], Lambert *et al.* discretize the plan with a multi-resolution grid and then route the edges on that grid. To avoid twisty routes and to respect the *Edge drawing constraint*, we modified the discretization step of [LBA10] to only use a quadtree. Figure 4 shows zoomed views on the same network bundled using the original algorithm of [LBA10] (a) and only using a quadtree (b). One can easily notice that using a quadtree reduces the number of bends per edge and respects the *Edge drawing constraint*.

3.5. Some results: visualizing yeast metabolic network

Figure 5 shows the resulting drawing of the yeast metabolic network provided by BioCyc database [KOMK*05]. That network contains 836 nodes (422 metabolites, 414 reactions) and 936 edges spread over 164 pathways.

One can notice that the biological conventions our algorithm tries to respect were correctly taken into account. Indeed no duplication has been performed, the representation is not cluttered, edges are drawn in a pseudo-orthogonal manner (except for inner cycle edges) and topological structures have been detected. Nevertheless the set of independent pathways computed during the partitioning step contains 33 pathways and many others were spread all over the drawing area. As mentioned in section 2, this is mainly due to the respect of the *Duplication constraint*. Our technique can overcome that problem by allowing the user to give a set of pathways for which the *Closeness constraint* will be strictly respected. For instance, if interested in the aerobic respiration, the user can decide to insert the *TCA cycle pathway* into the set of independent pathways as it was done to generate that picture. One can notice in the zoomed view (see Figure 5) that both the *Closeness constraint* and the *Topological constraint* were respected for the *TCA cycle pathway* as that pathway is drawn in a close area of the drawing and the longest cycle of reactions contained in that pathway (also known as Krebs cycle) has been detected and correctly represented.

3.6. Duplication constraint relaxation

In the previous sections, we described how our drawing technique can lay out the entire metabolic network while avoiding node duplication. However, when considering very dense and/or complex networks, the partitioning scheme may compute a small set of independent pathways leading to a representation in which numerous pathways have been spread all over the drawing area. Moreover, the produced visualization can also suffer from clutter issues. To overcome these problems, we propose to relax the *Duplication constraints* according to two levels of clutter:

- **Pathway level:** when a metabolite is involved in many reactions, then not duplicating that metabolite may create

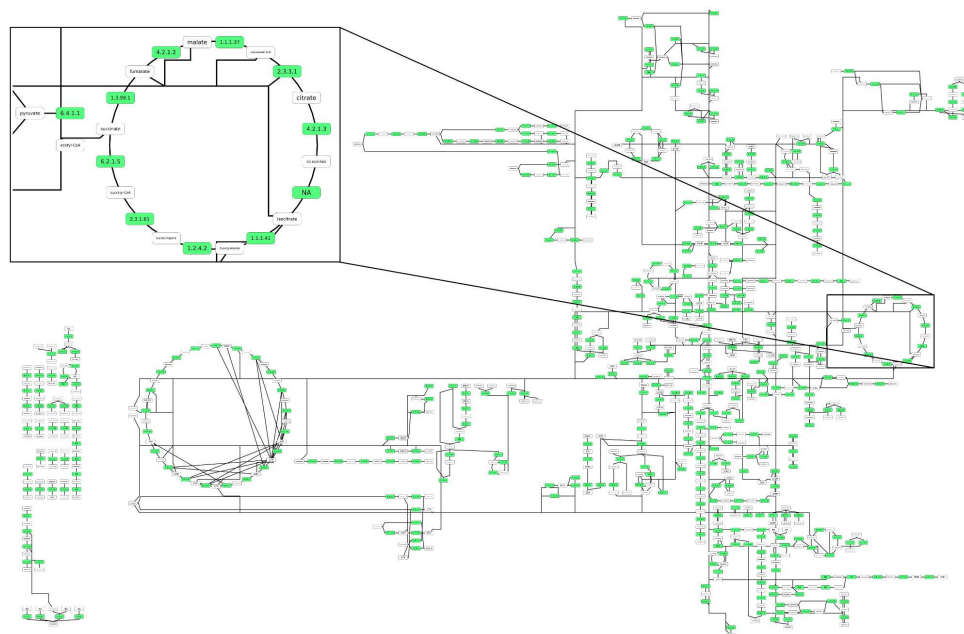


Figure 5: Yeast metabolic network drawn using our method and driving the set of independent pathways by adding the TCA cycle pathway (pathway responsible for aerobic respiration) in it. In the zoomed view, one can see that the cycle of reactions contained in the TCA cycle pathway has been correctly detected and represented.

an unnecessary large number of edge-crossings. To solve that issue, we provide the user with the ability to duplicate metabolites according to the number of reactions it is involved in.

- **Network level:** when some pathways share metabolites and/or reactions with many other pathways, then the computation of the set of independent pathways may produce poor results. We also provide the user with the ability to duplicate metabolites and reactions according to the number of pathways containing them.

For instance, let us consider the yeast metabolic network described in section 3.5. When computing the set of independent pathways, our algorithm provides a set containing 33 pathways, i.e., 20% of all pathways. If duplicating all the reactions and metabolites belonging to more than 3 different pathways, then our algorithm computes a set of 86 independent pathways, i.e., 52% of the set of pathways. Our technique can thus guarantee that the resulting representation will strictly respect the *Closeness constraint* for 86 pathways (instead of 33) for that network. The representation of the network after relaxing the *Duplication constraint* contains less clutter than the one we can compute without any duplication. This approach can be seen as a compromise between the *Duplication constraint* and the *Closeness constraint*. Figure 6 shows visualizations of yeast metabolic network (a) after the duplication of nodes belonging to more than 3 pathways and (b) after the duplication of all nodes.

In that Figure, pathways have been surrounded by concave hulls (see section 4).

3.7. Time complexity vs computation times

In term of time complexity, the bottleneck of our framework lies in the partitioning step as the problem of finding the longest cycle has been proved to be NP-Complete [Kar72] and as we use an exact method to detect them. Therefore if the input network only contains one metabolic pathway, the longest cycle detection algorithm is applied to the whole network. This leads to an overall time complexity $O(exp(n))$.

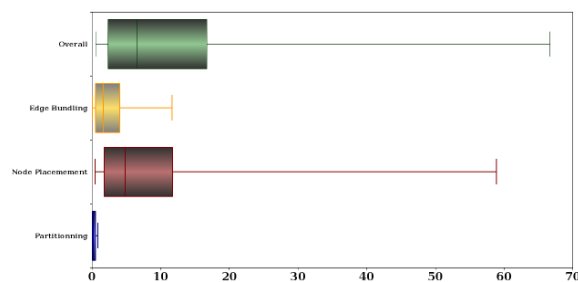


Figure 7: Boxplots of computation times in seconds for the 113 organisms of MetExplore DataBaBase [CWV*10].

Nevertheless that worst case never happens in real-world

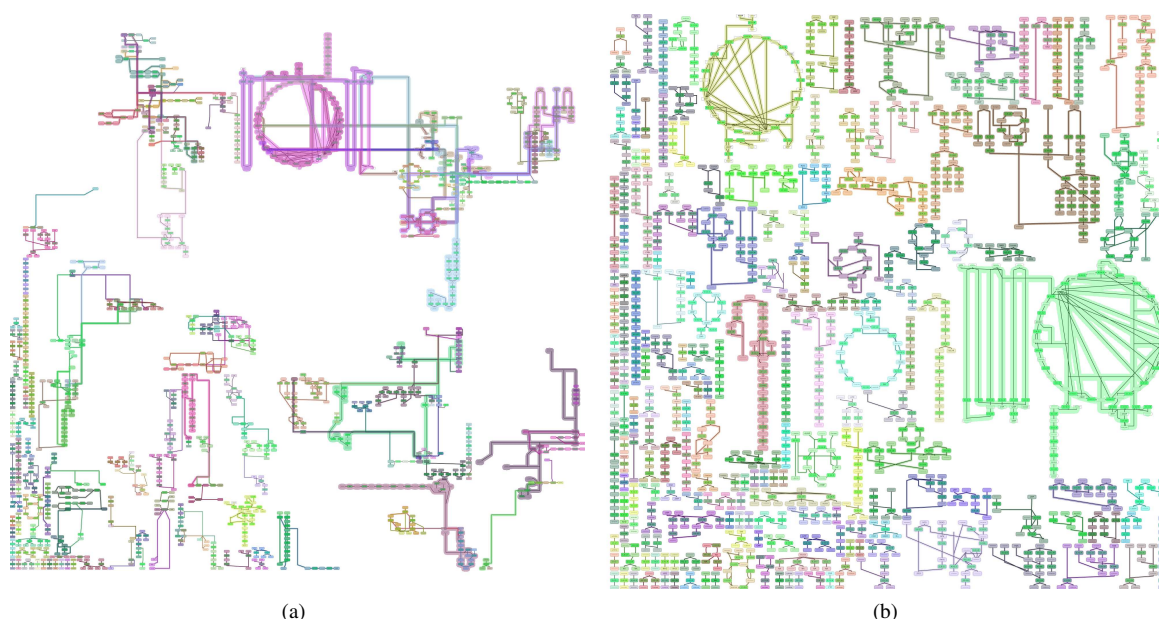


Figure 6: Yeast metabolic network (a) after duplication of nodes belonging to more than 3 pathways -204 nodes have been duplicated- and (b) after duplication of all nodes -835 nodes have been duplicated and one is not belonging to any pathway. Pathways have been surrounded by concave hulls (see section 4)

scenarios and our framework can draw a network within few seconds. Figure 7 shows the boxplots of the computation times in seconds of our algorithm for all the organisms of MetExplore DataBase [CWV*10]. That database contains the metabolic networks of 113 different organisms, from simple ones like *Candidatus Hodgkinia cicadicola* (144 nodes, 133 edges and 21 pathways) to more complex ones like *Homo sapiens* (1932 nodes, 2140 edges and 272 pathways).

The computation times of the whole framework vary from 0.5 seconds (for *Candidatus Hodgkinia cicadicola*) to 66.7 seconds (for *Mycobacterium smegmatis*) and the average computation time (resp. the median) is 11.3 seconds (resp. 6.5 seconds). One can also notice in figure 7 that the computation time of the partitioning step is very low (the average computation time is 0.33 and the median is 0.27) in comparison to the overall computation time. While the longest cycle detection is the bottleneck of the time complexity, it seems that the node placement step is more time consuming (the average computation time is 8.7 and the median is 4.8).

4. Visualizing pathways in the global network context

Even if the produced representation respects the network topology, the edge bundling phase makes the task of identifying individual metabolic pathways hard. Indeed, similar edges are grouped together and can share several bends along their polyline shape. Due to the ambiguity induced by

the numerous bundles crossroads, following a single edge from its source node to its target node may be difficult. To overcome that problem we introduce two complementary focus+context interaction techniques which aim at retrieving relevant pathway information. Starting from a network element selected by the user: metabolite or reaction, those techniques emphasize the different pathways containing it. The first one helps to identify the concerned pathways location in the layout by adding concave hulls around them. The second one aims at focusing on each pathway and getting better insight of their topological structure by applying a 3D distortion to network elements.

4.1. Emphasizing pathways with concave hulls

Being able to retrieve pathway information relative to a particular metabolite or reaction is a real needed feature to any metabolic network visualization system. The simplest technique should be coloring each focused pathway but visualizing their intersections is not easily feasible that way. So we preferred to surround pathways of interest with concave hulls. In that manner, users can easily locate and identify requested pathways but also find shared elements between them by looking at the hulls intersection. Figure 10(a) illustrates that interaction technique.

To generate the concave hulls, we use an image-based version of the technique recently introduced by Collins *et al.* [CPC09]. First, each subgraph to be emphasized is ren-

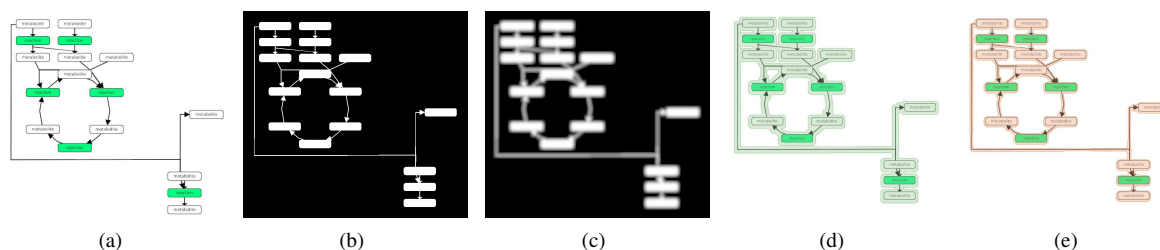


Figure 8: Illustration of the concave hulls generation process. (a) Subgraph pattern to emphasize. (b) Off-screen rendering of the pattern with all colors set to white. (c) Normalized scalar field obtained by convolving (b) with a Gaussian kernel. (d) Hulls extracted with a threshold of 0.1 when running Marching Square on (c). (e) Hulls extracted with a threshold of 0.5 when running Marching Square on (c).

dered in an off-screen framebuffer by setting all graph elements colors to white (Figure 8(b) illustrates that step). Then, a Gaussian convolution is performed on the generated image and the result is normalized. This way, we obtain a scalar field on which we can extract iso-surfaces (see Figure 8(c)). Pixel values vary according to their distance to the graph-skeleton from 1.0 (when the distance to the skeleton is 0) to 0.0 (when the distance is larger than a given threshold). The size and the spread of that field are determined by the radius and the standard deviation of the Gaussian kernel. Next, subgraph contours are extracted by running a two dimensional version of the well-known *Marching Cube* algorithm [LC87] (Figure 8(d) and (e) show hulls obtained using different parameters for [LC87]). Finally, concave polygons points in scene coordinates are computed from the pixel coordinates of the extracted hulls.

4.2. Focus+context interaction for uncovering pathway information

Emphasizing pathways with concave hulls helps users to locate them in the metabolic network representation. However, due to the network size and the edge-bundled layout, some ambiguities can remain when one wants to focus on a particular pathway. For instance, when a pathway is laid out on a large area of the drawing, it can sometimes be hard to determine if a node lying in its hull really belongs to it. To overcome these issues, we introduce a 3D distortion based interaction technique which aims at highlighting a subgraph pattern in the global network context.

Our technique is close to the Graph Folding of Carpendale *et al.* [CCFS95] with the difference that it distorts the view only at the pattern local scale. The principle is to modify the z-coordinate of the graph elements to render and use a perspective projection to visualize the distortion effect. The area of the graph layout in which the deformation will be applied is determined by the pattern skeleton and the distortion radius. The pattern skeleton is defined by a set of points and segments. In our case, the pattern is a subgraph

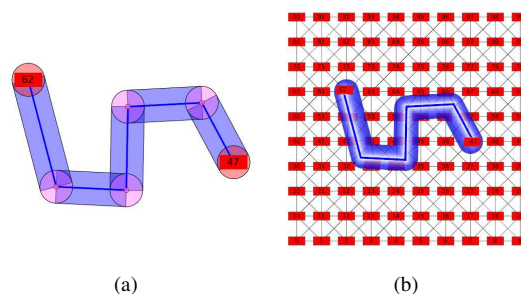


Figure 9: Illustrations of the focus+context technique used to emphasize a subgraph pattern in a network. (a) A simple subgraph extracted from a network with two nodes and one edge. The area on which the 3D deformation will be applied is represented. (b) Result of the technique when focusing on the subgraph introduced in (a).

of a metabolic network, so the points correspond to the nodes and edges bends coordinates of the graph and the segments correspond to the edges ones. For each vertex of the geometric primitives used to represent the network elements, the distance to the pattern skeleton components having an influence on the vertex is computed. A skeleton point has an influence on a vertex if the distance between them is less than the distortion radius. A segment has an influence on a vertex if the orthogonal projection of the vertex on the straight line defined by the segment lies on it and if the distance between the vertex and its projection is less than the distortion radius. The minimum distance is then retained to modify the z-coordinate of the vertex by applying a drop-off function. In our case, we use a simple hemispherical drop-off function $f(x) = 1 - x^2$, $0 \leq x \leq 1$ so that the pattern to emphasize will be fold over a cylinder surface. Other types of drop-off function can be employed to achieve different levels of magnification as explained in [CLP04]. This simple algorithm has been implemented using OpenGL vertex shaders which

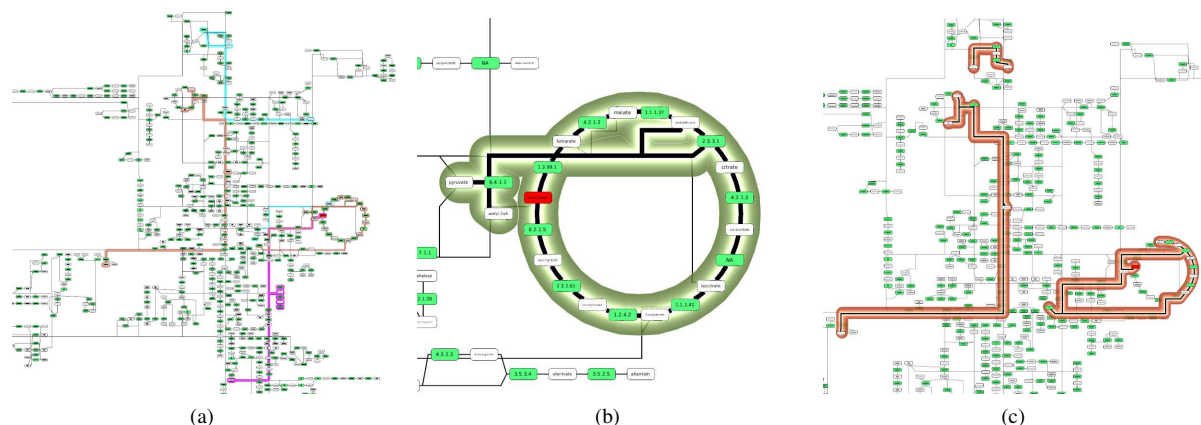


Figure 10: Retrieving pathways information using dedicated focus+context interaction techniques. (a) By selecting an element in the network, all pathways containing it are surrounded by concave hulls. (b) and (c) Focusing on a single pathway containing the selected element (colored in red) using a 3D deformation when rendering the network.

applies the 3D deformation in real time when rendering the network.

To increase pathway magnification in the global context, we also reduce clutter within the pattern influence area by applying an inverse deformation (a negative z-coordinate is used) to other elements located in that area. Moreover, to help users to clearly perceive the magnified pathway, we render a shaded surface obeying the same deformation used to emphasize the pattern. Illustrations of that technique on a sample network can be found in Figure 9. The results we obtained when applying it to magnify pathways in our produced metabolic network representation are introduced at Figures 10(b) and 10(c).

5. Conclusion

In this paper, we have presented a new drawing technique to lay out the entire metabolic network while preserving the pathway information as much as possible. That technique combines partitioning, node placement and edge bundling to provide a pseudo-orthogonal visualization. A key feature of our method is to enable the user to decide whether the most important information lies in the network topology (by prohibiting node duplication) or in its decomposition into pathways.

We also introduce complementary focus+context interaction tools that emphasize relevant pathways in the entire metabolic context. This is particularly helpful to retrieve pathway information as edges overlaps in bundled visualization can lead to misinterpretations. The first one helps to identify location of focused pathways in the layout by surrounding them with concave hulls. The second one aims at focusing on a particular pathway and getting better insight

of its topological structure by applying a 3D distortion to network elements.

An interesting direction for future work is to explore different methods to define whether or not duplication should be performed but also which nodes should be duplicated. This could be achieved by using dedicated stress function (such as in force-directed algorithms) that could predict how node duplication could reduce the clutter of the representation. Another direction is to take advantage of the user's knowledge to determine if a node should be duplicated. According to his understanding of the biological process, but also according to the visualization, the expert will decide which duplication will improve the drawing readability while being biologically meaningful. Finally, we plan to conduct an experimental study on the visualization of network patterns in a global context in order to evaluate the effectiveness and efficiencies of our interaction tools.

References

- [Aub03] AUBER D.: Tulip : A huge graph visualisation framework. In *Graph Drawing Softwares*, Mutzel P., Jünger M., (Eds.), Mathematics and Visualization. Springer-Verlag, 2003, pp. 105–126. 1, 4
- [BDS03] BRANDES U., DWYER T., SCHREIBER F.: Visualizing Related Metabolic Pathways in Two and Half Dimensions. In *Proc. Graph Drawing 2003 (GD'03)* (2003), pp. 110–122. 1
- [BLC*07] BOURQUI R., LACROIX V., COTTRET L., AUBER D., MARY P., SAGOT M.-F., JOURDAN F.: Metabolic network visualization eliminating node redundancy and preserving metabolic pathways. *BMC Systems Biology* 1 (2007), 29. 2, 3
- [Bou08] BOURQUI R.: Décomposition et Visualisation de graphes: Applications aux Données Biologiques, 2008. 3
- [BR01] BECKER M., ROJAS I.: A graph layout algorithm for drawing metabolic pathways. *Bioinformatics* 17 (2001), 461–467. 1

- [CCFS95] CARPENDALE M. S. T., COWPERTHWAIT D. J., FRACCHIA F. D., SHERMER T.: Graph folding: Extending detail and context viewing into a tool for subgraph comparisons. In *Proceedings of the Symposium on Graph Drawing (GD'95)*, volume 1027 of *Lecture Notes in Computer Science* (1995), Springer, pp. 127–139. 8
- [CLP04] CARPENDALE S., LIGH J., PATTISON E.: Achieving higher magnification in context. In *Proceedings of the 17th annual ACM symposium on User interface software and technology* (New York, NY, USA, 2004), UIST '04, ACM, pp. 71–80. 8
- [CPC09] COLLINS C., PENN G., CARPENDALE M. S. T.: Bubble sets: Revealing set relations with isocontours over existing visualizations. *IEEE Trans. Vis. Comput. Graph.* 15, 6 (2009), 1009–1016. 7
- [CWV*10] COTTRET L., WILDRIDGE D., VINSON F., BARRETT M. P., CHARLES H., SAGOT M.-F., JOURDAN F.: MetExplore: a web server to link metabolomic experiments and genome-scale metabolic networks. *Nucleic Acids Research* 38, suppl 2 (jul 2010), W132–W137. 6, 7
- [DEG*05] DOGRUSOZ, ERSON, GIRAL, DEMIR, BABUR, CETINTAS, COLAK: Patikaweb: a web interface for analyzing biological pathways through advanced querying and visualization. *Bioinformatics* 22, 3 (November 2005), 374–375. 1
- [EF96] EADES P., FENG Q.-W.: Multilevel Visualization of Clustered Graphs. In *Proc. Graph Drawing, GD* (Berlin, Germany, 18-20 1996), no. 1190, Springer-Verlag, pp. 101–112. 3
- [FLM94] FRICK A., LUDWIG A., MEHLDAU H.: A Fast Adaptive Layout Algorithm for Undirected Graphs. In *Proc. Graph Drawing 1994 (GD'94)* (1994), pp. 388–403. 5
- [GJ83] GAREY M. R., JOHNSON D. S.: Crossing Number is NP-Complete. *SIAM Journal on Algebraic and Discrete Methods* 4, 3 (1983), 312–316. 3
- [GZ05] GABOUJE S. D., ZIMÁNYI E.: Generic visualization of biochemical networks: A new compound graph layout algorithm. In *Poster Proc. of the 4th International Workshop on Efficient and Experimental Algorithms (WEA 05)* (2005). 1
- [JTG*05] JOSHI-TOPE G., GILLESPIE M., VASTRIK I., D'EUSTACHIO P., SCHMIDT E., DE BONO B., JASSAL B., GOPINATH G. R., WU G. R., MATTHEWS L., LEWIS S., BIRNEY E., STEIN L.: Reactome: a knowledgebase of biological pathways. *Nucleic Acids Research* 33 (2005), D428–D432. 2
- [Kar72] KARP R.: Reducibility Among Combinatorial Problems. In *Complexity of Computer Computations*, Miller R. E., Thatcher J. W., (Eds.). Plenum Press, 1972, pp. 85–103. 3, 6
- [KCVG*05] KESELER I., COLLADO-VIDES J., GAMACASTRO S., INGRAHAM J., PALEY S., PAULSEN I., PERALTAGIL M., KARP. P.: Ecocyc: A comprehensive database resource for *Escherichia coli*. *Nucleic Acids Research* 33 (2005), D334–D337. 1
- [KGF*10] KANEHISA M., GOTO S., FURUMICHI M., TANABE M., HIRAKAWA M.: KEGG for representation and analysis of molecular networks involving diseases and drugs. *Nucleic Acids Research* 38 (2010), D355–D360. 1
- [KOMK*05] KARP P., OUZOUNIS C., MOORE-KOCHLACS C., GOLDOVSKY L., KAIPA P., AHREN D., TSOKA S., DARZENTAS N., KUNIN V., LOPEZ-BIGAS N.: Expansion of the BioCyc collection of pathway/genome databases to 160 genomes. *Nucleic Acids Research* 19 (2005), 6083–6089. 1, 5
- [LBA10] LAMBERT A., BOURQUI R., AUBER D.: Winding Roads: Routing edges into bundles. *Computer Graphics Forum* 29, 3 (2010), 853–862. 5
- [LC87] LORENSEN W. E., CLINE H. E.: Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1987), SIGGRAPH '87, ACM, pp. 163–169. 8
- [LCTS08] LACROIX V., COTTRET L., THÉBAULT P., SAGOT M.-F.: An introduction to metabolic networks and their structural analysis. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 99, 1 (2008). 1
- [Mic93] MICHAL G.: *Biochemical Pathways* (Poster). Boehringer Mannheim, 1993. 2
- [Mic98] MICHAL G.: On representation of metabolic pathways. *BioSystems* 47 (1998), 1–7. 2
- [MWS*10] MEYER M., WONG B., STYCZYNSKI M., MUNZNER T., PFISTER H.: Pathline: A Tool for Comparative Functional Genomics. *Computer Graphics Forum* 29 (2010). 1
- [Nic97] NICHOLSON D.: Metabolic pathways map (poster), 1997. Sigma Chemical Co., St. Louis. 2
- [PK06] PAILEY S., KARP P.: The Pathway Tools cellular overview diagram and Omics Viewer. *Nucleic Acids Research* 34, 13 (august 2006), 3771–3778. 2
- [RHR*10] ROHRSCHEIDER M., HEINE C., REICHENBACH A., KERREN A., SCHEUERMANN G.: A Novel Grid-Based Visualization Approach for Metabolic Networks with Advanced Focus&Context View. In *Proc. Graph Drawing 2009 (GD'09)* (2010), pp. 268–279. 2
- [SAA09] SIMONETTO P., AUBER D., ARCHAMBAULT D.: Fully Automatic Visualisation of Overlapping Sets. *Computer Graphics Forum* 28, 3 (2009), 967–974. 3
- [SBM] Sbm viewer. <http://sbw.kgi.edu/layout/>. 2
- [Sch03] SCHREIBER F.: Comparison of metabolic pathways using constraint graph drawing. In *APBC 03: Proceedings of the First Asia-Pacific bioinformatics conference on Bioinformatics* (2003), Australian Computer Society, Inc., pp. 105–110. 1
- [SMO*03] SHANNON P., MARKIEL A., OZIERAND O., BALIGA N., WANG J., RAMAGE D., AMIN N., SCHWIKOWSKI B., IDEKER T.: Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks. *Genome Research* 13 (2003), 2498–2504. 2
- [SND05] SARAIYA P., NORTH C., DUCA K.: Visualizing biological pathways: requirements analysis, systems evaluation and research agenda. *Information Visualization* 4 (2005), 1–15. 2
- [SSE*02] SIRAVA M., SCHÄFER T., EIGLSPERGER M., KAUFMANN M., KOHLBACHER O., BORNBERG-BAUER E., LENHOF H.-P.: BioMiner - modeling, analyzing, and visualizing biochemical pathways and networks. *Bioinformatics* 18 (2002). 1
- [ST04] SYMEONIDIS A., TOLLIS I. G.: Visualization of biological information with circular drawings. In *Intl Symposium on Medical Data Analysis (ISBMDA)* (2004), pp. 468–478. 5
- [STT81] SUGIYAMA K., TAGAWA S., TODA M.: Methods for visual understanding of hierarchical system structures. *IEEE Transactions on Systems, Man, and Cybernetics* 11, 2 (1981), 109–125. 1
- [vWGW02] VAN HELDEN J., WERNISCH L., GILBERT D., WODAK S.: Graph-based analysis of metabolic networks. *Ernst Schering Research Foundation Workshop* 38 (2002), 245–274. 3
- [WK05] WEGNER, KUMMER: A new dynamical layout algorithm for complex biochemical reaction networks. *BMC Bioinformatics* 6 (2005), 212. 1