BIOINFORMATICS ORIGINAL PAPER Vol. 27 no. 16 2011, pages 2248–2255
doi:10.1093/bioinformatics/btr351

Systems biology Advance Access publication June 11, 2011

MGV: a generic graph viewer for comparative omics data

Stephan Symons* and Kay Nieselt

Center for Bioinformatics Tübingen, Faculty of Science, University of Tübingen, 72076 Tübingen, Germany

Associate Editor: Martin Bishop

ABSTRACT

Motivation: High-throughput transcriptomics, proteomics and metabolomics methods have revolutionized our knowledge of biological systems. To gain knowledge from comparative omics studies, strong data integration and visualization features are required. Knowledge gained from these studies is often available in the form of graphs, and their visualization is especially useful in a wide range of systems biology topics, including pathway analysis, interaction networks or gene models. Especially, it is necessary to compare biological models with measured data. This allows the identification of new models and new insights into existing ones.

Results: We present MGV, a versatile generic graph viewer for multiomics data. MGV is integrated into Mayday (Battke *et al.*, 2010). It extends Mayday's visual analytics capabilities by integrating a wide range of biological models, high-throughput data and meta information to display enriched graphs that combine data and models. A wide range of tools is available for visualization of nodes, data-aware graph layout as well as automatic and manual aggregation and refinement of the data. We show the usefulness of MGV applied to several problems, including differential expression of alternative transcripts, transcription factor interaction, cross-study clustering comparison and integration of transcriptomics and metabolomics data for pathway analysis.

Availability: MGV is a open-source software implemented in Java and freely available as a part of Mayday at http://www.microarray-analysis.org/mayday.

Contact: symons@informatik.uni-tuebingen.de

Supplementary information: Supplementary data are available at Bioinformatics online.

Received on May 2, 2011; revised on June 6, 2011; accepted on June 8, 2011

1 INTRODUCTION

The current focus of life science research is to achieve a systems-based view of processes, organisms and ecosystems. To this end, biological activity is studied using a variety of tools: gene expression on transcription level is measured using microarrays and next-generation sequencing. Furthermore, gas chromatography and mass spectrometry methods are applied to measure protein and metabolite concentrations. This leads to large and complex datasets containing potentially tens of thousands of measured species from hundreds of experiments. These data are then used to build a comprehensive view of biological processes, with exhaustive formal models of the systems as one of the ultimate aims. Analysis and interpretation of such data call for visualization at every step to be successful. Visualization tools like box plots and heat maps support the choice of methods for quality control and normalization, as well as during statistical analysis. The identification of relationships in the data and hypothesis generation often require graph-based visualizations. This includes biological pathways, regulatory and interaction networks, as well as gene models.

The full extent of systems biology data is only utilizable if analysis tools can keep up with the data. Current datasets require classical visualization tools, a genome-based view as well as strong network visualization features. Especially for the latter, smart layout and data analysis tools as well as interactivity is necessary and extensibility is important to cope with new data formats and analysis methods. Network visualization is most useful if it is combined with measured data, as this allows to compare predictions with actual results. Therefore, extensible tools for visualizing and analyzing enriched networks are desirable. Additionally, the integrative bioinformatics paradigm calls for seamless integration and processing of data from a multitude of sources.

Here, we introduce a generic, integrative graph viewer called MGV (short for Mayday Graph Viewer). MGV is based on the versatile Mayday platform (Battke *et al.*, 2010) and offers a comprehensive set of tools for analysis and visualization of graphs. As an extensible, feature rich (Koschmieder *et al.*, 2011) tool for multiple omics data analysis, Mayday is flexible about the data analyzed. A recently published major extension to Mayday, called Mayday SeaSight (Battke and Nieselt, 2011), allows working with high-throughput sequencing data. MGV is designed to work on any kind of measured data, and it can import different graph and pathway data formats. A large variety of different configurable tools are available for displaying measured data at nodes, along with layout methods that use data properties of the nodes. Several biological and all-purpose graph file formats can be imported. MGV also allows to visually organize and analyze biological data. To this end, we investigated new ways of exploring, organizing and summarizing systems biology data in graphs.

While the extent of systems biology data expands, so does our knowledge of the organisms, processes and molecules under investigation. Much of that knowledge is condensed in databases of biological pathways. As a conceptual representation, they are essential for understanding data and putting it into context: pathways are the building blocks of our understanding of life at its basis. Several common sources of biological pathways are available, for example KEGG (Kanehisa *et al.*, 2008), MetaCyc (Caspi *et al.*, 2008), Pathway Commons (Cerami *et al.*, 2011), WikiPathways (Pico *et al.*, 2008) or Reactome (Matthews *et al.*, 2009). One of the most useful concepts for working with pathways is a graph representation. It allows to view the components

*To whom correspondence should be addressed.

of the pathway that are interacting in close context. For example, it is natural to view metabolites as nodes and reactions as edges. A petri network representation of pathways [see for example Pinney *et al.* (2003)] allows better display of regulatory processes and is standardized in SBGN (Le Novère *et al.*, 2009), the Systems Biology Graphical Notation. In general, exploratory analysis of pathways can lead to more effective identification of hypotheses (Kelder *et al.*, 2010). Graphs are also a natural representation of regulatory networks and protein interaction data. Also gene models have been successfully represented as graphs (Heber *et al.*, 2002), and so have overlapping sets (e.g. clusters of (co)regulated genes) of any origin.

There are several tools available for the visualization of pathways and general biological networks. These include generic tools like Vanted (Junker *et al.*, 2006), Ondex (Köhler *et al.*, 2006) and Cytoscape (Smoot *et al.*, 2011), a popular platform for the analysis of networks with more than 100 plugins for data analysis. However, most of these tools have few data analysis tools for the underlying biological data and instead focus on graph visualization. For the field of biological pathways, many customized solutions for specific aspects and organisms exist; among others KaPPA-View (Tokimatsu *et al.*, 2005), MapMan (Thimm *et al.*, 2004) and Paintomics (García-Alcalde *et al.*, 2010). GenMapp (Salomonis *et al.*, 2007) and PathVisio (Van Iersel *et al.*, 2008) provide rich pathway visualizations for a wide range of organisms. In contrast, a plethora of methods for all kinds of analyses is implemented in R (R Development Core Team, 2008), mostly based on the Bioconductor (Gentleman *et al.*, 2004) project. While this is immensely useful and also offers a large number of visualization plots, R does not offer much interactivity in general.

All these tools have interesting methods, but often require researchers to exchange data between applications, which is time consuming and often causes unnecessary problems like incompatible formats, loss of data during conversion and repeated analyses. Furthermore, most applications do not make full use of the potential of the graph representation. Other applications are limited to certain organisms, data sources or cannot integrate measured species data. However, graphs, especially when displaying additional information on vertices and edges, can be used to visualize many aspects of data, also simultaneously in detail and as overviews. A graph-based view of measured data is also a helpful tool for manual or semisupervised structuring of the data.

With MGV, we intend to provide an integrated solution for all graph-based data visualization, with the special focus on measured data integration and visualization. Based on Mayday, MGV is ensured to work on many high-throughput datasets without conversion overhead or need for third-party applications.

2 METHODS

This section introduces the MGV, and its most important components, for node rendering, layout and data integration, based on its architectural design.

2.1 Graph-based visualization of data

In Mayday, measured species, independent of the source, are called 'probes'. In Mayday's data model, probes can be manually or automatically grouped to unsorted sets. Meta information, for example gene annotations, quality values or genetic loci, can be associated with any component of the data model. The conceptual framework for MGV is a digraph $G=(V,E)$, which might be unconnected or even a degenerate case with $E=\emptyset$. Each node $v \in V$ can either be one or more probes representing experimental data or metadata, such as molecules representing reaction components in biochemical pathways. This concept allows to visualize data from different studies. Furthermore, both nodes and edges are associated with a name, a set of properties and a role which describes their biological meaning. The user is presented an embedding of G in the plane, which can be interactively modified or changed using several algorithms. The graphical representation of nodes and edges depends on their respective roles and is configurable. MGV is interconnected with all other visualization plots available in Mayday [see Battke *et al.* (2010) for an overview], allowing to display any perspective of the data. Furthermore, selections are shared, as are data transformations. Any set $B \subseteq V \neq \emptyset$ can be visually grouped in MGV. Groups can be displayed either in a rectangular or elliptical shape or as the convex hull of the nodes as embedded on the screen. Possible operations on groups are the calculation and display of distance and correlation heat maps. Visualization of probe values and lists of probe properties or the most frequent annotations can be displayed for the groups (see Supplementary Fig. 1 for examples). Groups can either be built manually or based on node and probe properties, for example all nodes with probes can be grouped according to the major gene lists of the probes. Also, k-means and Qt clustering (Heyer *et al.*, 1999) can be applied to induce groups, if nodes are associated with probes. Modularity clustering (Newman, 2006) is also available (implementation from Noack, 2007) for group creation.

New graphs in MGV can be created from nodes originating from probes or sets of probes, between which edges can be introduced manually or automatically. Further nodes can be added manually or from existing Mayday data. Interaction partners can be queried from STRING (Szklarczyk *et al.*, 2010). External annotations can be imported from PubMed and UniProt.. For the creation of other graphs, MGV contains an extensible collection of tools to produce new graphs for several purposes. These include, among others, clustering comparison, biochemical pathways, gene models and a probe-centric view (useful to gather information about a singe gene), all discussed below.

Partitioning clustering, significance tests and similar methods applied to two or more different studies usually lead to different results. MGV provides a graphical comparison tool for clusterings from two studies. Clusters are represented as nodes in a bipartite graph, their intersection as edges connecting the clusters. This allows to inspect how clusters are shared between the two studies. If the total set of probes used in two datasets are identical, a P-value can be calculated using a hypergeometric test (Fury *et al.*, 2006). Small or insignificant overlaps can be filtered out.

MGV can import biochemical pathways from KEGG and BioPax files. For KEGG pathways, the layout as defined in the file is used. BioPax pathways are displayed using SBGN. Additionally, overview graphs of BioPax files, induced by either reactions or pathways as nodes and the metabolites they share as edges, can also be created. The import of graphs from SBML and CellML files is also possible.

The analysis of differential splicing calls for a genome-based view that shows the different transcripts of a gene. MGV can visualize differential splicing along with expression levels of transcripts as measured by tiling micorarrays or RNAseq. This is done either based on splicing graphs (Heber *et al.*, 2002) or in a verbose style representing each isoform as a linked list. For an intermediate view, a tree-based structure can be used that aggregates common 5′-end exons. Exon-level data can also be viewed in MGV.

For a single probe, MGV can be used to aggregate all available information and display it in a star-like graph. Information considered encompasses the probe selections the target probe is contained in, the profile of the probe in other datasets, meta information, genomic neighbors, abstracts from PubMed, interaction partners from STRING, as well as reaction partners from BioPax files. This view can help researchers to swiftly learn about specific genes (see Supplementary Fig. 2 for an example).

Networks from GML, XGMML, GraphML and GraphViz dot files can be imported. Networks can be saved in GraphML, Dot and GML formats or a specialized format (based on GraphML) that preserves all properties of the current graph. When importing graphs, probes are automatically mapped

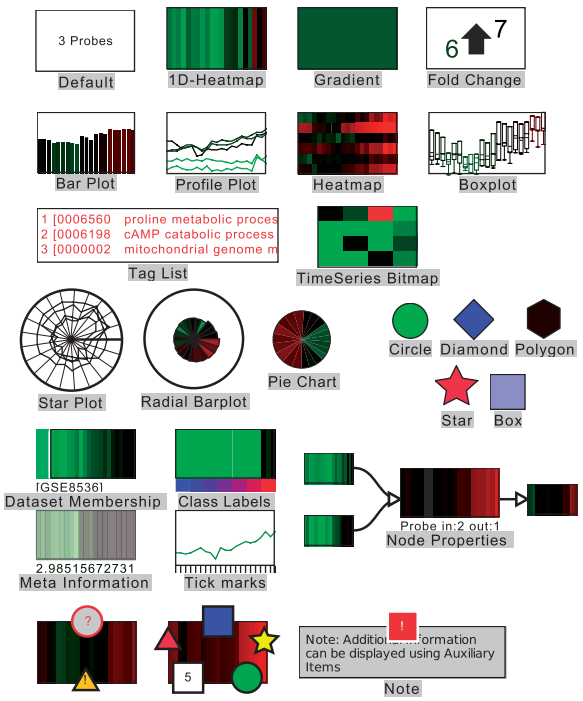

Fig. 1. Rendering options of MGV. Plots for single values and single or multiple genes are available. Simple representation of nodes is done with various shapes. Additional information can be displayed via renderer decorators and auxiliary items.

toall imported nodes when possible. Export of images is possible in PDF, SVG and several bitmap formats.

Automatic extension of the graphs is possible by adding similar probes (with respect to one of several distance measures). If a mapping of probes to genomic positions is available, genomic neighbors of probes can be added. External information, such as PubMed abstracts and interaction partners (via STRING) can be added to complement the knowledge of the researcher. Finally, nodes carrying a dynamic average of all probes connected to the node can be added, for example to summarize the activity of a class of enzymes. Several ways of weighting incoming and outgoing edges are available. To remove unnecessary nodes from the graph, filtering can be done on all aspects of nodes. Nodes can also be merged to form nodes with several probes.

Implementation: MGV can be extended via plugins, including data import, node rendering and layout, as well as graph manipulation and filtering. MGV, as an extension of Mayday, is implemented in Java and both are free and open-source software licensed under the GPL (version 2), available at http://www.microarray-analysis.org/mayday.

2.2 Node rendering

MGV associates every node and edge with a role, which are used to decide how the node or edge is displayed. For each node, the rendering can be dynamically configured using a wide range of display options. A large choice of renderers is available (Fig. 1). If a node is associated with one or more probes, the probe values are displayed using either color gradients, heat maps, profile plots, star plots, bar plots and (for a large number of probes) box plots. Various shapes are available for showing the node's class or properties. Special renderers are available for components like exons (in gene models) and SBGN entities. For the summary of long time series, a time series bitmap (Kumar *et al.*, 2005) renderer is available.

In addition to the primary renderers described above, additional information can be displayed using so-called decorators. Decorators are placed below or above a node. They are used to display the origin (set of probes or dataset) or meta information connected with this node, for example class partition, gene ontology annotations and relevance values derived from statistical tests. Further additional information can be added using auxiliary items. They are displayed as symbols placed on the margin of the node. Auxiliary items can be used to manually add node-wise additional information and can express uncertainty, highlight important facts and mark additional aspects.

Edges with different roles mostly differ in source and target decorations (i.e. arrowheads) or line style. Edges can be drawn in a tapered style (Holten and Wijk, 2009) to display their direction, which is considered more readable than arrowheads. For displaying the weight of the edges, the width of the edge can be adjusted or the edge can be displayed in a zigzag, with the frequency encoding the weight. Reduction of edge overdrawing can be achieved by using Bezier curves, thus heuristically bundling edges. In addition to several predefined roles, it is possible to define additional roles of edges and nodes for specialized rendering. Furthermore, each node and edge can be configured to be rendered in an individual style.

2.3 Graph layout

For general graphs with a non-empty set of edges, standard layout algorithms (hierarchical, force-based, etc.) are available. For graphs with no edges, these algorithms are not applicable. Instead, such graphs are laid out on grids or on circles. If probes are associated with nodes, their properties and values can be used to induce an embedding, for example using principal component analysis to place similar probes near each other.

Grouping and sorting nodes according to various criteria is used to distribute groups of similar nodes on the screen. This can be done either in rectangular, circular or axis-wise fashion (the latter as introduced as Hive Plot, http://mkweb.bcgsc.ca/linnet/). Criteria for grouping include node properties, node connectivity, probe values and probe annotation. Nodes can be sorted according to several criteria within their group in order to reproducibly structure the nodes within a component. This kind of procedure allows to introduce external knowledge into the layout. This is also applicable to general graphs, where for example (strongly) connected components could form groups. An advantage of this method is the runtime, which is $O(g \cdot n \log n)$ for n nodes and g groups. Pathways are drawn in a recursive scheme, see Symons *et al.* (2010) for details. In addition to the layout of the entire graph, users may automatically rearrange parts of the graph, by aligning them or centering entire connected components around a node in a hierarchical way.

2.4 Cross dataset operations

Working across several datasets is an important feature in MGV. In systems biology, for example, transcriptomic, proteomic and metabolomic data measure different aspects of a biological system. Usually, such data cannot be directly compared, which means they cannot be combined in the same dataset. However, visual comparison allows important insights and can be done using MGV. Since the graph data structure is agnostic about the origin of the probes, probes from different datasets in Mayday can be represented in a common graph. To ensure comparability, a probe mapping keeps track of which probes are equivalent between datasets. The same is done for equivalent experiments. The various data transformations available in Mayday are communicated between datasets, as well as selections of mapped probes. Only a subset of the plugins is applicable to nodes with probes of different origin, which makes visualization the most important tool for this data. For visualization of probe data on nodes, no probe or experiment mapping is required. Therefore, MGV can be used to visually compare any data. However, statistical analyses between datasets require a valid probe and experiment mapping. Based on grouped nodes of different datasets, distance and correlation (Pearson, Spearman, Kendall) measures can be calculated. A correlation heat map allows to inspect correlation between probes from two

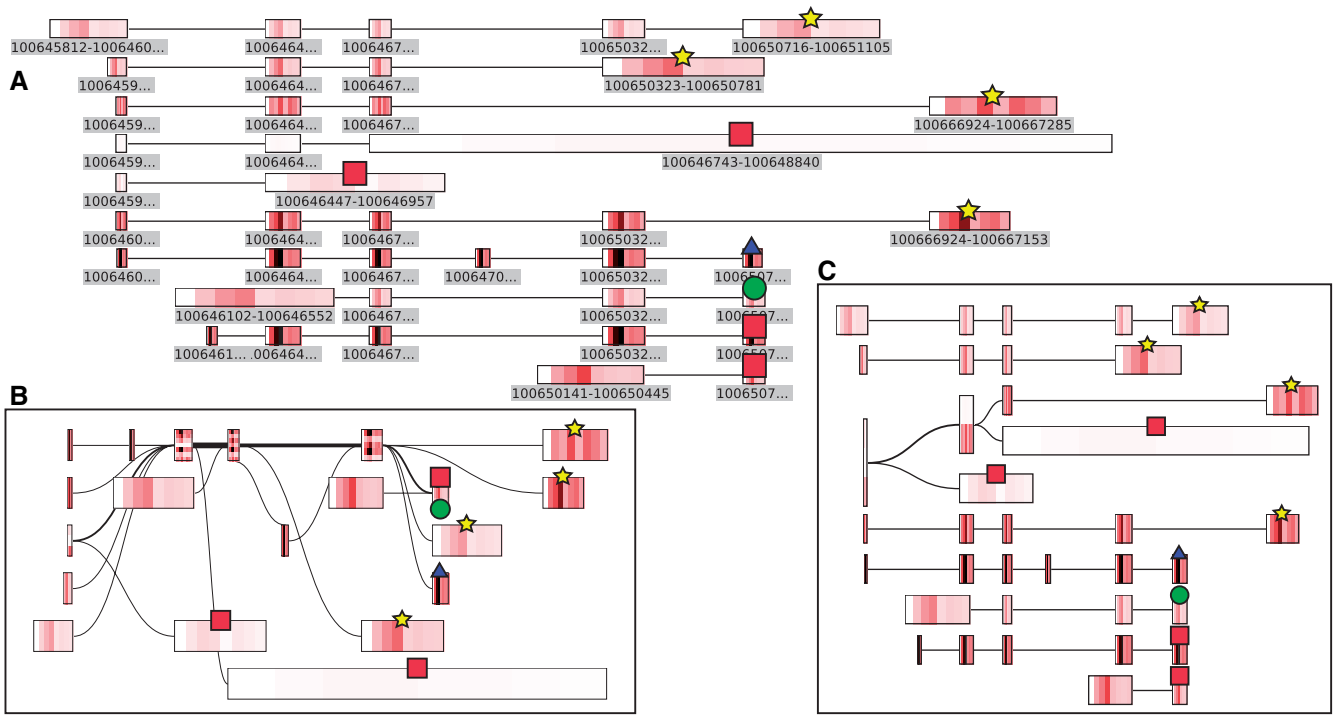

Fig. 2. Differential expression within and between isoforms of *RPL36A* in human. The 3′-most exons are manually labeled according to the Ensembl biotype: square: retained intron; triangle: nonsense medicated decay; circle: processed transcript; star: protein coding. Expression is visualized in a heat map style, with an inverse black body radiation gradient. (black: high; white: low). (**A**) Shows a verbose view of the gene. The labels below the nodes indicate the genomic position of the exon; tool tips show the complete label. (**B** and **C**) Show the simplified and compressed view, respectively.

datasets, along with their respective values (see Supplementary Fig. 3 for an example).

3 RESULTS

We illustrate some of the most important features of MGV using typical questions in systems biology research. Since there are different data requirements for each question, we use a variety of datasets for illustration. In all examples, we use measured values from expression and/or metabolomics studies mapped to nodes. To visualize the expression values, we employ the inverse black body radiation gradient, which works well in gray scale and color. Since demonstrating every feature of MGV exceeds the scope of this paper, the Supplemental Material contains several further examples.

3.1 Differential expression of gene isoforms

Using RNAseq or tiling microarrays, it is possible to measure gene expression on the gene isoform level. MGV can visualize both differential expression within and between isoforms of the same gene. In Figure 2, we show an example of human gene *RPL36A* (ribosomal protein L36a) that has 10 isoforms. The expression values were generated from RNAseq data (Illumina, Inc., unpublished data). Marked with a triangle, an isoform annotated in Ensembl to be subject to nonsense medicated decay and a similar one, annotated as a retained intron biotype (marked with a square), show distinct upregulation in some experiments. In comparison, three of the four protein coding isoforms have a much lower expression in these experiments. The remaining transcripts with retained introns have very low expression values across all experiments. In addition to the expression profiles of the transcripts, MGV can show the overall structure of the transcripts. In Figure 2B, identical exons in the 5′ end of the transcripts are aggregated, which allows to focus on the 3′ ends of the transcripts. Only three isoforms of *RPL36A* share common exons at their 5′ end. The splicing graph in Figure 2C shows that there are some shorter exons in the center of the gene that occur in many transcripts. They are indicated by the high number and weight of adjacent edges. In this case, expression profiles are displayed using a heat map with multiple lines.

3.2 Transcription factor activity in yeast

We applied MGV to two publicly available datasets studying transcription in *Saccharomyces cerevisiae* under various conditions. Knijnenburg *et al.* (2007) studied yeast under four nutrient limited conditions in chemostat cultures, in both aerobic and anaerobic conditions. Each condition was investigated in triplicates. In total, the dataset consists of 24 experiments. Marks *et al.* (2008) investigated yeast during wine fermentation in seven time points in triplicates. The entire study has 21 experiments. Both studies were conducted using the Affymetrix Yeast Genome S98 platform, which has 9335 probesets. The normalized datasets provided by GEO (accession IDs GSE1723 and GSE8536) were directly imported into Mayday.

The authors of the chemostat study identified several so-called modules, intersecting clusters of co-regulated genes and a set of

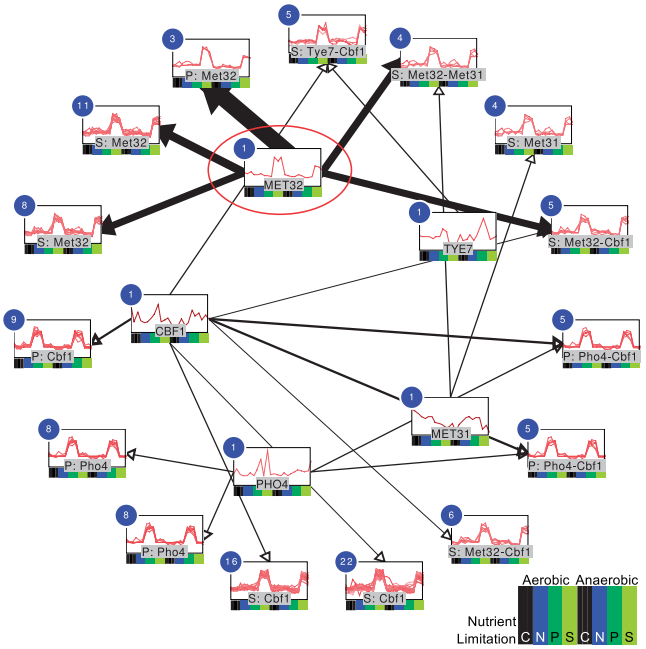

Fig. 3. Enriched visualization of transcription factors (inner ring of nodes) regulating sets of genes (outer ring) in yeast. Expression values are displayed as profile plots. Edge width is proportional to the mean correlation between transcription factor and target gene expression. The number of genes represented by each node is shown in the upper left corner of the node. The nutrient limitations for carbon (C), nitrogen (N), phosphorous (P) and sulfur (S) under aerobic and anaerobic conditions are displayed as a colored bar below the plot.

transcription factors that potentially control the modules. For a subset of each module, specific transcription factor (TF) binding sites were found. These annotations induce a directed graph of 70 nodes (of which Fig. 3 shows a subset), representing either TFs or sets of genes controlled by the TFs, and 59 edges, where an edge is drawn if a set of genes is controlled by a TF. The complete graph has 13 connected components and can be found in Supplementary Figs 4 and 5. Expression values either of the transcription factors or the controlled genes are mapped to the nodes. We added meta information for class labels and number of probes for each node and adjusted rendering rules to show profile plots for all nodes. Edge weights were calculated as the average Pearson correlation between the probes represented by the nodes adjacent to the edge. The graph was laid out using the grouping-based method: sources (transcription factors, inner ring) and sinks (target genes, outer ring) were grouped respectively. The largest component (20 nodes, representing altogether 44 unique probes) is displayed in Figure 3. From this figure, we can infer that *Met32* has a high correlation with several modules of genes that are upregulated in sulfur-limited conditions. *Pho4* has a similar, but weaker correlation to genes regulated by phosphorous limitation. The other transcription factors have much weaker correlation with their respective target genes than *Met32* and *Pho4*.

3.3 Clustering comparison

It can be suspected that there is an overlap of the genes regulated by nutrient depletion or anaerobic conditions identified in the chemostat study with genes regulated during the fermentation time series. We wish to answer the following questions: (1) how are genes which are differently regulated during wine fermentation regulated under nutrient or oxygen limitation? (2) How do genes controlled by nutrient limitation react during wine fermentation?

We performed *t*-tests to identify genes differently regulated by nutrient depletion and oxygen supply. For each condition, the expression differences against all other samples were investigated. For the wine study, tests were performed for all time points against the first time point. In both cases, we considered genes with $P < 0.05$ (after correction to control the false discovery rate) to be significant. For both studies, the resulting genes were clustered using *k*-means, with $k = 10$ and $k = 12$, respectively.

Using the clustering comparison tool of MGV, a visual comparison of the resulting genes was performed (Fig. 4). To do so, a graph was created, in which nodes represented clusters and were connected to clusters in the other dataset if they shared at least one probe. The edge weight was set to the overlapping fraction of the size of the smaller cluster. Nodes with no adjacent edges were omitted from the graph. Nodes were ordered descendingly by the weight of all adjacent edges. In Figure 4, the nodes adjacent to edges representing 10 and 20% overlap are shown separately. Concerning the first question, we can conclude from Figure 4 that especially genes upregulated during the early phase of the fermentation (especially during the first 48 h) overlap with genes upregulated in carbon-depleted conditions, regardless of oxygen supply. They also overlap with a cluster of 62 genes mainly upregulated in anaerobic carbon- and phosphorus-limited conditions. Furthermore, we find that genes downregulated at the beginning of the fermentation (1 h) overlap with genes downregulated in aerobic conditions. To a lower extent, this is also true for a cluster of 60 genes downregulated toward the end of the fermentation. Concerning question two, we see that nutrient depletion regulated genes, except for carbon, have little overlap with genes regulated during fermentation. Limited intersection of genes upregulated under sulfur and of genes upregulated under nitrogen limitation exists with genes downregulated only during the beginning of the fermentation, while nitrogen-related genes are also upregulated after 48 h in the fermentation study. Phosphorous-regulated genes share only very small intersections with genes regulated during fermentation. When directly comparing the differentially regulated genes (see Supplementary Fig. 6), similar conclusions can be made. In addition, a set of genes downregulated under nitrogen limitation is upregulated at the beginning of the fermentation.

3.4 Cross-dataset visualization of pathways

We demonstrate the cross-dataset visualization features of MGV with a dataset studying anaerobic growth in *Pseudomonas aeruginosa* (Alvarez-Ortega and Harwood, 2007). Microarray gene expression data (ArrayExpress accession id E-GEOD-6741) was imported into Mayday. Here, we found that highly variant genes (coefficient of variation > 0.1) contained, among others, genes from amino acid degradation pathways.

To complement this data, we added similar but not directly comparable metabolomics data from the Systomonas project (Choi *et al.*, 2007) that studied *P.aeruginosa* under similar conditions (series 6 and 8). Figure 5A shows the tyrosine degradation pathway of *P.aeruginosa* in MGV as an SBGN process diagram, with enzyme

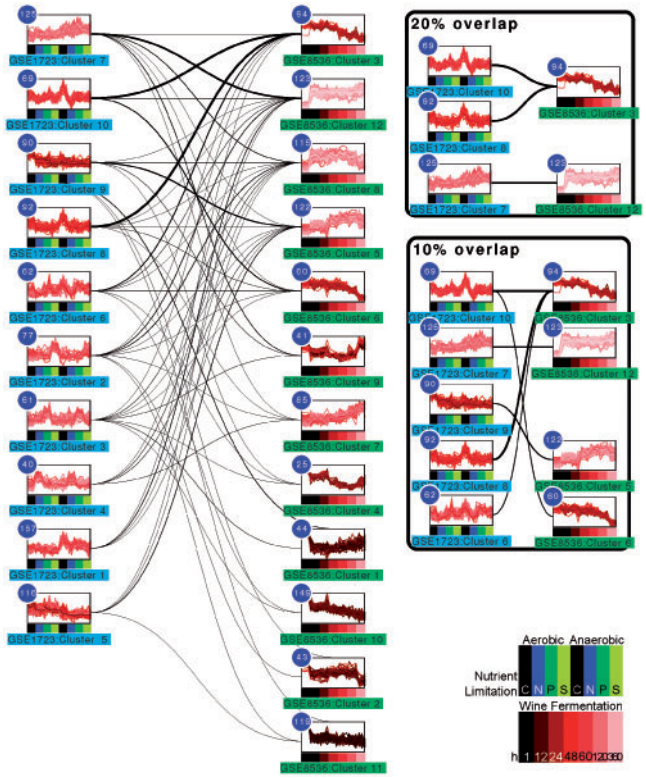

Fig. 4. Bipartite cluster comparison graph: a comparison of clustered differentially expressed genes between the chemostat study [Knijnenburg *et al.* (2007); left] and the wine study [Marks *et al.* (2008); right]. Clustering was performed with k-means with $k=10$ and $k=12$ clusters, respectively. Expression is visualized as a profile plot. Edge weight is proportional to the overlapping fraction of the smaller cluster. Intersections of more than 20 and 10%, respectively, are shown separately on the right.

activity estimated by transcript expression from the microarray data and metabolite abundance estimated from the metabolomics study. Figure 5B shows an alternative view of the enzymes in the tyrosine degradation pathway. Especially the first enzyme, branched-chain amino acid amino transferase (PA5013) is involved in several similar reactions. The reduced transcription of this enzyme under aerobic conditions correlates with the reduced amino acids concentration.

3.5 Performance

MGV has moderate memory requirements. The two datasets used for the yeast studies, encompassing 9335 genes and 45 experiments in total, along with extensive meta information require 280 MB. All examples shown here (including supplements) required no more than 50 MB additional memory. Memory consumption grows linearly with the number of graph objects. Gene profiles shown at a node do not contribute toward the memory consumption significantly. Rendering speed mostly depends on the number of probes and edges in a graph, and the renderers used.

4 DISCUSSION

In this article, we have introduced MGV, a new versatile and extensible graph viewer for systems biology data. We combined the successfully implemented concepts of graph-based visualization of biological knowledge with the concept of using small multiples (Tufte, 1983) for visualization of quantitative data. In order to demonstrate the feasibility of our implementation of this concept, we used data from different sources and contexts and applied MGV to investigate common questions on it.

For regulatory networks, MGV provides an integrated view of the network and of the expression data, in different levels of detail. In general, the levels of detail can easily be configured via the many rendering options for nodes. Possible details range from zero values (showing only the shape of the node) to several hundred values (via heat maps and profile plots). The latter case can suffer from overplotting when many probes are displayed. In this case, summaries of the probes, e.g. box plots can be used. This is an application of the visual analytics concept: data are summarized (or omitted, e.g. by filtering out nodes or edges) in order to identify interesting regions in the graph. Then the level of details can be vastly increased, by zooming in and adding meta information to the nodes.

Versatility is a design goal for MGV also in the context of data integration. MGV brings together quantitative data with annotation data and textbook knowledge. The integration of expression data with gene models allows to simultaneously visualize two levels of differential activity: splicing and transcription, which is especially interesting in larger RNAseq studies, where different classes of samples are compared. The most useful rendering strategy for exon nodes are heat maps and profile plots. Circular plots are less useful here, as it is necessary to keep the width of the node constant.

MGV can work on a wide variety of data and is able to join data between different sources and studies. This encompasses several studies at the same level, e.g. two microarray studies as well as the comparison of transcriptomics (or proteomics) with metabolomics studies. We have illustrated this with the *P.aeruginosa* data, in the context of a metabolic pathway in the BioPax format, giving both an overview and details of the process. A direct comparison of the metabolomics and transcriptomics data though is not possible, since the studies differ in several criteria. Still, these studies are comparable enough to demonstrate the use of MGV for this application. Furthermore, to our knowledge, there is no comprehensive multiomics dataset publicly available with whole genome transcription data and a large number of measured and identified metabolites. As technology advances, such datasets will eventually be made available and MGV is well suited to work on them, as demonstrated.

A further cross-study visualization feature is the clustering comparison tool of MGV. It allows to investigate overlaps of probe groupings between datasets. In this way, properties of the probes in each partition can be compared, as in the chemostat and wine fermentation data. This allowed us to identify overlaps between fermentation-related genes and nutrient- or oxygen-dependent genes, without using external annotations. Further applications include comparing profiles of groups in closely related studies and investigating the stability of clusterings among studies.

MGV not only assists in exploratory analysis of pathways, but can also help in formulating new and confirmed hypothesis. For this purpose, new graphs can be easily built in MGV. The resulting graphs can be used to illustrate such a hypothesis, which can be then combined with the underlying data in an interactive environment.

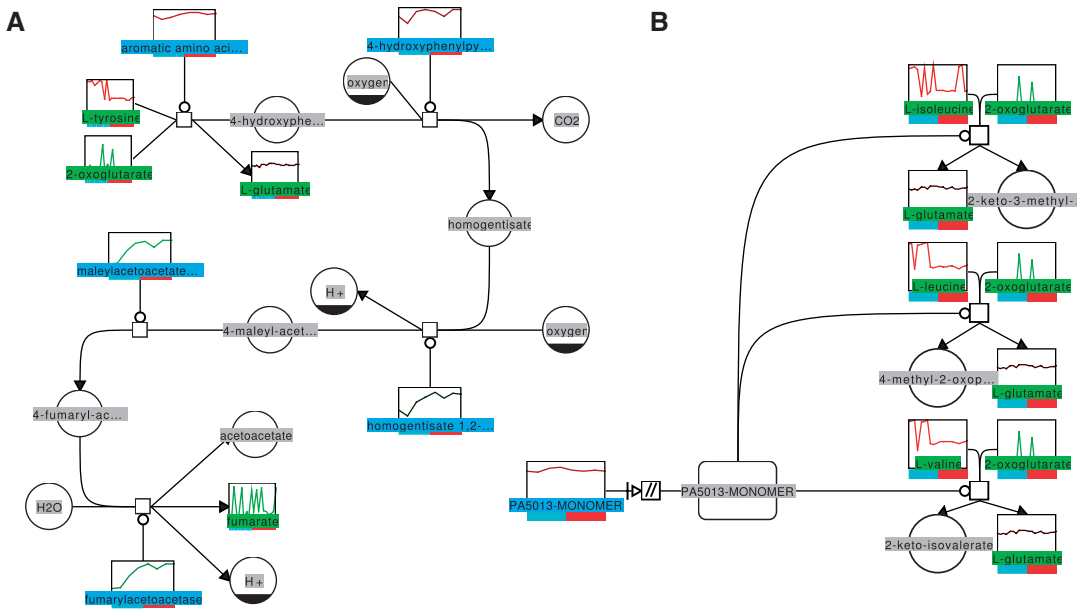

Fig. 5. Visualization of pathways in MGV. (**A**) Shows the tyrosine degradation pathway of *Pseudomonas aeruginosa* in SBGN format, with profile plots for both microarray-based transcriptomics data and metabolomics data from different studies. The colors of the labels denotes the origin of the data. (blue: transcriptomics, green: metabolomics). (**B**) Shows an automatically generated schematic overview of reactions catalyzed by the branched-chain amino acid aminotransferase (PA5013). The same data as in (A) is used. Throughout the figure, class labels are displayed as colored bars below the plot (blue: anaerobic; red: aerobic conditions).

Based on Mayday, MGV is integrated into a well-established framework, connected with many analysis and visualization tools. We consider this to be beneficial because the workflow of data analysis requires several data integration and analysis steps before the high level analyses and visualizations that MGV is designed for. New methods emerge continuously. Therefore, we consider it to be important that all key features of MGV, node rendering, graph layout, data import, graph manipulation and filtering, are extensible. We are continuously extending MGV to provide new methods for all these purposes. While extensibility and breadth of functionality were in the focus of development, MGV is able to process large sparse graphs with thousands of genes.

Further research is necessary on the questions of graph drawing and visualization of dense datasets, especially in the context of how to use the data properties of a node. For the latter topic, a thorough user study should evaluate assets and liabilities of the current and future approaches. Enhancements of the rendering options and speed are also planned. Cross-study comparison methods are also of increasing interest. A large variety of methods for data integration has been proposed, some of which require careful review and integration into MGV. The development of MGV will continue in these directions. For graph drawing, methods as suggested by Stajdohar *et al.* (2010) for large graphs or Adai *et al.* (2004) for handling unconnected graphs can greatly enhance MGV. Further extensions in usability, e.g. a scripting feature for automation and performance enhancements are planned.

5 CONCLUSION

MGV provides a set of powerful tools to integrate and visualize systems biology data from many sources. High-dimensional data visualization in a graph context is a powerful method to integrate data from all omics sources with meta information and external knowledge. MGV provides a wide range of tools for this purpose. We have shown examples of data from genomes, transcriptomics and metabolomics, which were seamlessly integrated and visualized using MGV, even across datasets. Graph layout methods that use the data at the nodes further enhance the analysis of data. With upcoming new multiomics studies, MGV will be a useful tool to make the most out of this data.

ACKNOWLEDGEMENT

We acknowledge Florian Battke for helpful discussions and his extensive work on data integration and normalization in Mayday and Alexander Herbig for helpful input and discussions.

Conflict of Interest: none declared.

REFERENCES

Adai,A.T. *et al.* (2004) LGL: creating a map of protein function with an algorithm for visualizing very large biological networks. *J. Mol. Biol.*, **340**, 179–190.

Alvarez-Ortega,C. and Harwood,C. (2007) Responses of Pseudomonas aeruginosa to low oxygen indicate that growth in the cystic fibrosis lung is by aerobic respiration. *Mol. Microbiol.*, **65**, 153.

Battke,F. and Nieselt,K. (2011) Mayday SeaSight: combined analysis of deep sequencing and microarray data. *PLoS One*, **6**, e16345.

Battke,F. *et al.* (2010) Mayday–integrative analytics for expression data. *BMC Bioinformatics*, **11**, 121.

Caspi,R. *et al.* (2008) The metacyc database of metabolic pathways and enzymes and the biocyc collection of pathway/genome databases. *Nucleic Acids Res.*, **36**, D623–D631.

Cerami,E.G. *et al.* (2011) Pathway commons, a web resource for biological pathway data. *Nucleic Acids Res.*, **39**, D685–D690.

Choi,C. *et al.* (2007) SYSTOMONAS–an integrated database for systems biology analysis of Pseudomonas. *Nucleic Acids Res.*, **35**, D533.

Fury,W. *et al.* (2006) Overlapping probabilities of top ranking gene lists, hypergeometric distribution, and stringency of gene selection criterion. In *Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE*, Vol. 1, IEEE, New York, NY, USA, pp. 5531–5534.

García-Alcalde,F. *et al.* (2010) Paintomics: a web based tool for the joint visualization of transcriptomics and metabolomics data. *Bioinformatics* , **27**, 137–139.

Gentleman,R. *et al.* (2004) Bioconductor: open software development for computational biology and bioinformatics. *Genome Biol.*, **5**, R80.

Heber,S. *et al.* (2002) Splicing graphs and EST assembly problem. *Bioinformatics* , **18** (Suppl. 1), S181–S188.

Heyer,L.J. *et al.* (1999) Exploring expression data: identification and analysis of coexpressed genes. *Genome Res.*, **9**, 1106–1115.

Holten,D. and Wijk,J.V. (2009) A user study on visualizing directed edges in graphs. *Proceedings of the 27th international conference on Human factors in computing systems (CIH 2009)*, ACM, New York, NY, USA, p. 2299.

Junker,B.H. *et al.* (2006) VANTED: a system for advanced data analysis and visualization in the context of biological networks. *BMC Bioinformatics*, **7**, 109.

Kanehisa,M. *et al.* (2008) Kegg for linking genomes to life and the environment. *Nucleic Acids Res*., **36**, D480–D484.

Kelder,T. *et al.* (2010) Finding the right questions: exploratory pathway analysis to enhance biological discovery in large datasets. *PLoS Biol.*, **8**, e1000472.

Knijnenburg,T.A. *et al.* (2007) Exploiting combinatorial cultivation conditions to infer transcriptional regulation. *BMC Genomics*, **8**, 25.

Köhler,J. *et al.* (2006) Graph-based analysis and visualization of experimental results with ONDEX. *Bioinformatics* , **22**, 1383–1390.

Koschmieder,A. *et al.* (2011) Tools for managing and analyzing microarray data. *Brief. Bioinformatics* [Epub ahead of print, doi: 10.1093/bib/bbr010].

Kumar,N. *et al.* (2005) Time-series Bitmaps: a practical visualization tool for working with large time series databases. In *SIAM 2005 Data Mining Conference*, SIAM, Philadelphia, PA, USA, pp. 531–535.

Le Novère,N. *et al.* (2009) The systems biology graphical notation. *Nat. Biotechnol.*, **27**, 735–741.

Marks,V.D. *et al.* (2008) Dynamics of the yeast transcriptome during wine fermentation reveals a novel fermentation stress response. *FEMS Yeast Res.*, **8**, 35–52.

Matthews,L. *et al.* (2009) Reactome knowledgebase of human biological pathways and processes. *Nucleic Acids Res.*, **37**, D619–D622.

Newman,M.E.J. (2006) Modularity and community structure in networks. *Proc. Natl Acad. Sci. USA*, **103**, 8577–8582.

Noack,A. (2007) Energy models for graph clustering. *J. Graph Algorithms Appl.*, **11**, 453–480.

Pico,A.R. *et al.* (2008) WikiPathways: pathway editing for the people. *PLoS Biol.*, **6**, e184.

Pinney,J.W. *et al.* (2003) Petri Net representations in systems biology. *Biochem. Soc. Trans.*, **31**(Pt 6), 1513–1515.

R Development Core Team (2008) *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Salomonis,N. *et al.* (2007) Genmapp 2: new features and resources for pathway analysis. *BMC Bioinformatics*, **8**, 217.

Smoot,M.E. *et al.* (2011) Cytoscape 2.8: new features for data integration and network visualization. *Bioinformatics*, **27**, 431–432.

Stajdohar,M. *et al.* (2010) FragViz: visualization of fragmented networks. *BMC Bioinformatics*, **11**, 475.

Symons,S. *et al.* (2010) Integrative systems biology visualization with MAYDAY. *J. Integr. Bioinformatics*, **7**.

Szklarczyk,D. *et al.* (2010) The STRING database in 2011: functional interaction networks of proteins, globally integrated and scored. *Nucleic Acids Res.*, **39**(Suppl. 1), D561–D568.

Thimm,O. *et al.* (2004) Mapman: a user-driven tool to display genomics data sets onto diagrams of metabolic pathways and other biological processes. *Plant J.*, **37**, 914–939.

Tokimatsu,T. *et al.* (2005) KaPPA-View. A web-based analysis tool for integration of transcript and metabolite data on plant metabolic pathway maps. *Plant Physiol.*, **138**, 1289.

Tufte,E.R. (1983) *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT.

Van Iersel,M. *et al.* (2008) Presenting and exploring biological pathways with PathVisio. *BMC bioinformatics*, **9**, 399.