# Interactive Visualization of Provenance Graphs for Reproducible Biomedical Research

Stefan Luger* Johannes Kepler University (JKU) Linz
Holger Stitz† JKU Linz
Samuel Gratzl‡ JKU Linz
Nils Gehlenborg§ Harvard Medical School
Marc Streit¶ JKU Linz

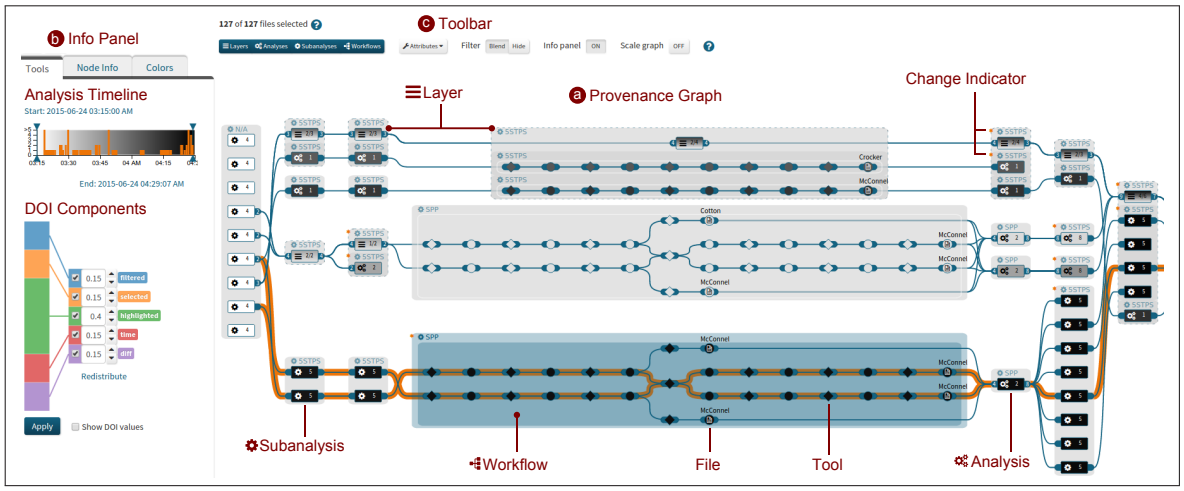


**Figure 1:** *The provenance graph (a) is aggregated and filtered based on the selected analysis execution time and the weighted degree-of-interest (DOI) components (b). In the top center of the graph, two horizontally aligned workflows show a compound layer node, where the top node represents the layer itself while two workflows are extracted based on their specific DOI value exceeding a predefined threshold. The toolbar (c) provides node type specific views (layer, analysis, subanalysis, workflow) and attribute mapping onto nodes.*

## Abstract

A major challenge of data-driven biomedical research lies in the collection and representation of provenance information to ensure reproducibility of the gained results. The *Refinery Platform* is an integrated data management, analysis, and visualization system designed to support reproducible biomedical research. In order to communicate and reproduce multi-step analyses on datasets that contain data of hundreds of samples, it is crucial to be able to visualize the provenance graph at different levels of detail. Most existing approaches are based on node-link diagrams, however, they usually do not scale well to large graphs. Our proposed visualization technique dynamically reduces the complexity of subgraphs through hierarchical aggregation and application of a degree-of-interest (DOI) function to each node. Triggered by user interactions, such as selecting a subset of analyses or a path in the graph, unaffected parts of the graph are dynamically aggregated into a glyph representation. We further reduce the complexity of the provenance graph visualization by layering identical or similar sequences of parallel analysis steps into an aggregated sequence.

**Keywords:** Time-varying graphs, modular degree-of-interest.

*e-mail: stefan.luger@jku.at
†e-mail: holger.stitz@jku.at
‡e-mail: samuel.gratzl@jku.at
§e-mail: nils@hms.harvard.edu
¶e-mail: marc.streit@jku.at

## 1 Introduction

The *Refinery Platform*[1] is an integrated data management and analysis platform that is designed to enable collaborative and reproducible biomedical research. Refinery handles data at the file level and facilitates the execution of workflows on one or more input files in the *Galaxy*[2] bioinformatics workbench. For each of these analyses, Refinery automatically tracks comprehensive provenance information, including workflows applied, workflow parameters, tool versions, input files, the user executing the analysis, and others. Every analysis consists of one or more subanalyses, which correspond to the execution of a Galaxy workflow on a set of input files. For example, if a user selects 10 files to be processed by a workflow that takes one input file and produces one output file per input file, then the corresponding analysis would have 10 inputs and 10 outputs and would consist of 10 subanalyses. Every analysis uses only exactly one workflow. Along with the meta data attributes that users can assign to files in Refinery, the provenance information represents a richly annotated graph that contains all information necessary to reproduce the findings of a study performed with the help of the system. The provenance graph contains different types of nodes, primarily file nodes and tool nodes, which represent the tools that were used to process the files in a particular workflow. In order to efficiently review, revise, or communicate how a study was conducted, a visual representation of the provenance graph is needed. Due to the dozens to hundreds of files in the datasets and the complexity of the workflows, provenance graphs often contain hundreds of nodes, making visualization challenging. Our goal is to develop an interactive provenance graph visualization that exploits the typical topology of provenance graphs in order to provide a representation that adjusts to the actions of the user.

[1] http://refinery-platform.org
[2] http://galaxyproject.org/

## 2 User Tasks and Requirements

Together with domain experts we elicited a series of tasks that need to be supported by an effective provenance graph visualization:

**T I: High-Level Overview** Analysts want to start the exploration by inspecting an aggregated version of the provenance graph, giving them a rough overview of which workflows were run how often, in which configuration, and at which point in time.

**T II: Attribute Encoding** Analyses are annotated with a series of attributes such as date and time of execution, in- and output files, and who triggered the analysis. This information needs to be encoded effectively using the basic visual channels such as color, shape, and size in combination with carefully designed glyphs.

**T III: Drill-Down on Demand** Although the graph should be presented as reduced as possible, analysts need to get down to the lowest level of detail. They should be able to interactively drill-down into subgraphs that are of current interest, while the rest of the graph should be kept in a compact representation as context.

**T IV: Investigate Changes** Changes can occur at the level of the input files, the workflows, and also its parameterization. The provenance visualization needs to provide the analysts with the means to explore, track, and understand the changes over time.

**T V: Investigate Causality** A crucial task in the exploration of provenance graphs is to let analysts investigate the chain of files and transformations that contributed to a certain analysis result. This task can be addressed by allowing analysts to highlight the full path through the graph that lead to one or more nodes of interest.

## 3 Related Work

Many of established workflow management systems, such as Galaxy, provide at least basic support for capturing and accessing provenance information. Several other, domain-independent tools have been developed that automatically track and visualize provenance information, such as the *Taverna*[3] and *Kepler*[4] scientific workflow systems. However, neither of these tools can be widely applied to large-scale data processing projects, such as cancer genomics studies, primarily due to scalability issues.

A widely known provenance visualization platform is *VisTrails* [2], in which a key concept is the visualization trail that stores provenance data about steps executed in the pipeline, making it possible to point out different parameter configurations over time.

However, all of the discussed provenance visualization solutions have in common that they cannot handle provenance graphs with more than a few dozen nodes while still being usable. From a visualization research point of view, provenance graphs comprise two major challenges: (1) they become large very quickly and (2) they contain time-dependent information. For both of these topics a large body of related work exists. The grand challenge is therefore to come up with an effective combination of existing techniques and strategies that lets analysts address the tasks formulated above.

## 4 Approach

The provenance graph visualization in Refinery is built around a dynamic node-link diagram that is enhanced with three concepts to address the tasks described in Section 2. Our approach dynamically decreases or increases the level of detail of the graph visualization based on the actions of the user to improve scalability. The visualization is implemented in JavaScript using the *D3* library[5]. We use the *DAGRE* library[6] for dynamic layout computations.

[3] http://www.taverna.org.uk/
[4] http://kepler-project.org/
[5] http://d3js.org/
[6] http://github.com/cpettitt/dagre/

Hierarchical Aggregation The hierarchical nature of the provenance graph can be exploited to incorporate semantic aggregation into the visualization approach. In Refinery, the provenance graph consists of analyses, which in turn consist of subanalyses that represent the execution of a workflow on a set of input files. Workflows are subgraphs that consist of atomic file and tool nodes. At each of the four levels (analysis, subanalysis, workflow, file/tool), a wide range of node attributes are encoded in the glyphs used to represent the nodes (T II). Glyphs and different levels of semantic aggregation are illustrated in Fig. 1 (a). The level of aggregation (T I, T III) can either be controlled manually by the user or automatically by a degree-of-interest function.

Layering Approach We use network motif discovery to aggregate similar analysis paths [4]. A motif is characterized by workflow type and parameters, subanalysis count, and in- and outgoing edges. Motifs are detected computationally and are aggregated into a compound layer node [5], effectively creating another hierarchy level above analyses.

Modular Degree-Of-Interest Function Our approach uses a modular degree-of-interest (DOI) function [1, 6] to determine the level of detail for every node, including aggregated nodes, of the provenance graph. The DOI function incorporates properties and topology of the graph (e.g., node relationships, node creation time) and actions taken by the user (e.g., selection, filtering, distance [3]). The DOI computed based on these criteria automatically controls the degree of hierarchical aggregation applied to the nodes, addressing T I, T III, T IV, and T V. High DOI values, for example, will be assigned to selected and highlighted components, triggering an expansion of the aggregate nodes or nodes along a highlighted path (Fig. 1 (a)). We use a stacked bar chart to represent the weight of the components that influence the computed DOI value (Fig. 1 (b)).

## 5 Conclusion and Future Work

In this poster we presented a provenance graph visualization for biomedical analysis pipelines in Refinery. We address the elicited tasks using a modular node-based DOI function for semantic zooming and a layering approach for aggregation of similar paths. In the future, we are planning to conduct a user study to investigate how to balance multiple competing aggregation strategies effectively.


## Acknowledgements

This work was funded by the Austrian Research Promotion Agency (FFG) (840232), the Austrian Science Fund (FWF) (P27975-NBL), and the US National Institutes of Health (K99 HG007583).


## References


[1] J. Abello, S. Hadlak, H. Schumann, and H.-J. Schulz. A Modular Degree-of-Interest Specification for the Visual Analysis of Large Dynamic Networks. *IEEE Transactions on Visualization and Computer Graphics*, 2013.

[2] J. Freire and C. T. Silva. Making Computations and Publications Reproducible with VisTrails. *Computing in Science & Engineering*, 14(4):18–25, 2012.

[3] G. W. Furnas. Generalized fisheye views. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '86)*, pages 16–23. ACM, 1986.

[4] E. Maguire, P. Rocca-Serra, S.-A. Sansone, J. Davies, and M. Chen. Visual Compression of Workflow Visualizations with Automated Detection of Macro Motifs. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2576–2585, 2013.

[5] H. Stitz, S. Gratzl, S. Luger, N. Gehlenborg, and M. Streit. Transparent layering for visualizing dynamic graphs using the flip book metaphor. In *Poster Compendium of the IEEE VIS Conference*. IEEE.

[6] F. van Ham and A. Perer. "Search, Show Context, Expand on Demand": Supporting Large Graph Exploration with Degree-of-Interest. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '09)*, 15(6):953–960, 2009.