Caleydo Web: An Integrated Visual Analysis Platform for Biomedical Data

Samuel Gratzl* Johannes Kepler University Linz
Nils Gehlenborg† Harvard Medical School
Alexander Lex‡ University of Utah
Hendrik Strobelt§ Harvard University
Christian Partl¶ Graz University of Technology
Marc Streit‖ Johannes Kepler University Linz

Abstract

In many science domains data analysis has replaced data acquisition, generation, and storage as the main challenge. This challenge stems not only from volume but also from complexity and heterogeneity of the data. Molecular biology is a prime example for this trend. We identified six key aspects that a visual analysis platform for biological data should support. We address these aspects with the development of *Caleydo Web*. In this poster we describe its architecture and give an example of how it can be used to create *StratomeX.js*, a sophisticated visualization technique for cancer subtype characterization.

Keywords: Visual analytics, web visualization, biological visualization, provenance, ID mapping, heterogeneous data.

1 Introduction

Significant breakthroughs in the acquisition but also in the storage of scientific data have shifted the grand challenge in many science domains to data analysis [7]. A prime example for this shift is molecular biology, where large initiatives like *The Cancer Genome Atlas* project and emerging technologies such as single cell gene sequencing produce vast amounts of heterogeneous data. Visual analysis is a key approach for making sense of the data. However, with datasets from different sources, with different meanings, on distinct levels of scale, and of various types (tables, text, graphs, etc.), there is the need for new visual analysis platforms that tackle these new challenges. In this poster we introduce and present the current state of *Caleydo Web*, a next generation open source visual analysis platform for biomedical data.

2 Key Aspects

From our experience with domain collaborators and designing visualization systems in the past [4, 5] we identified six key aspects that a visual analysis platform for biological data needs to support:

A I: Data Scale and Heterogeneity Not only is the size of datasets increasing, there is also a growing number of publicly available datasets that researches want to integrate. Taken together, we observe that the size, complexity, and heterogeneity increases beyond current analysis and visualization capabilities. The data spectrum ranges from clinical and expression data, over epigenetic data, to full genome sequence information. Challenges include accessing, processing, and interactively visualizing the data.

*e-mail: samuel.gratzl@jku.at
†e-mail: nils@hms.harvard.edu
‡e-mail: alex@sci.utah.edu
§e-mail: hstrobelt@seas.harvard.edu
¶e-mail: partl@icg.tugraz.at
‖e-mail: marc.streit@jku.at

A II: Identifier Management An important aspect when integrating datasets from various sources is the mapping of identifiers between different annotation systems (e.g., Entrez, DAVID). Mappings, however, can be 1:1, 1:*n*, *n*:*m*, or even more complex if they are based on partially overlapping gene locations. Also, entities of different types (e.g., gene, protein, samples) that can be defined on different levels of granularity (e,g., chromosome, gene, base pair) lead to additional challenges.

A III: Multiple Coordinated Views (MCV) The integrated analysis of multiple interconnected datasets can lead to new insights, yet it is often sensible to show different datasets as independent views, as the visualization can then be chosen to best represent the data. The coordination of these views provides the links between the datasets. The MCV system needs to visually link the entities across the annotation systems and granularity levels involved.

A IV: Provenance and Collaboration A recent review showed that it was not possible to reproduce the findings from almost 90% of over 50 cancer genomics studies [1]. This highlights the need for all stages of the analysis to be reproducible, interpretable, and communicable, including the visual analysis. Integrated support for provenance tracking, sharing of results, communication, and collaboration are essential.

A V: Integrated Data Analysis The integration of algorithms, statistics, and machine learning approaches like clustering or dimensionality reduction are crucial for most applications of visual analysis platforms to biomedical data. The back and forth between analysts and algorithms should be as tight and swift as possible. For instance, when a data query cannot provide immediate feedback due to the complexity of the query or the size of the data, the system should report intermediate results which the analyst can use to judge the correctness and suitability of the parametrization and adjust them if necessary [6]. Data mining algorithms can also be used for guiding analysts to interesting patterns proactively [8].

A VI: Adaptability The last key aspect deals with the adaptability to changing environments. A visualization framework needs to be flexible enough to allow for, e.g., the addition of new data types, storage backends, visualization techniques, or processing algorithms. The platform should also support the creation of customized setups that are tailored to a specific application use case.

3 Related Work

BioJS [3] is a library for representing biological data. Its core is a small event-driven architecture that can be extended via plugins that are collected in a public registry. Interfaces are not defined by the library but described within a plugin's documentation only. This allows easy setup and creation of plugins for a range of different data types (A VI and A I). However, developers aiming at using multiple plugins in a setup with multiple coordinated views have to handle the synchronization and data mapping between individual plugins manually—hampering A II and A III. Moreover, the library focuses on the visualization of data only, not how it is accessed or processed (A V). Dealing with large datasets in web-based frameworks is particularly challenging, since transferring the whole dataset to the client is not an option.

Caleydo [4] is a standalone visualization framework for biological data and the predecessor of the proposed framework. Caleydo supports A II and A III, however, it lacks support for large datasets (A I), since it is a client-only application in which all datasets are loaded into main memory. Moreover, it has only rudimentary support for provenance (A IV) via a simple undo mechanism and the integrated data processing (A V) is limited to a fixed set of hard coded algorithms, such as clustering algorithms.

4 Architecture

Caleydo Web is based on a client-server architecture with a plugin mechanism on both sides. Client and server are coupled loosely via REST and WebSocket interfaces such that individual components can be replaced. By default, a web browser-based client and a Python server are used. Alternative possible clients include an R client for using the server API as centralized data access, or server components written in different programming languages like Java.

The plugin architecture uses a runtime environment with lazy-loaded plugins implementing extensions on one or both ends. The types of extensions include visualizations, data providers, data types, data formatters, or applications. An application is a customized and specialized arrangement of plugins for a specific purpose. For example, *StratomeX.js*, is a web-based reimplementation of Caleydo *StratomeX* [8, 5]. Figure 1 illustrates the interplay between the individual components. We are also working on a public registry in which plugins can be published, explored, and shared.

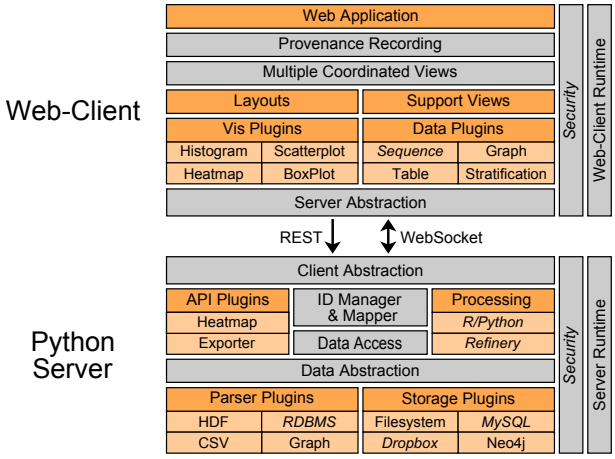

Figure 1: *Architecture of Caleydo Web. Italic labels indicate work in progress. Grey boxes represent Caleydo Web's core, dark orange boxes major plugin types, and light orange boxes custom plugins.*

This architecture allows to integrate all key aspects listed in Section 2. Large data (A I) can be handled by the web-client/server architecture. Depending on the data size, only partial, aggregated, or transformed data is transferred to the user. Mapping between different annotations (A II) is implemented using a graph database. Visualization plugins select items within their dataset and the platform takes care of converting the selection into their corresponding items in other visible datasets. By using a plugin-based approach, Caleydo Web is very flexible in terms of contained visualization techniques, dataset types, storages, and so on—addressing A VI. MCV setups (A III) are implemented by enforcing a minimal interface to visualization plugins including the location of individual data points. This allows the platform to create visual links across unknown representations. A command design pattern is used for managing provenance information (A IV). For the last aspect A V, we plan to use R, Python, and Refinery[1] for executing workflows. Intermediate results and feedback on the web-client are implemented using WebSocket communication.

[1] `http://www.refinery-platform.org/`

5 Implementation

Caleydo Web (`http://caleydo.org`) is open source under the BSD license and hosted on `https://github.com/Caleydo`. The client runtime of Caleydo Web is implemented in TypeScript and JavaScript using HTML5. This allows visualization plugin developers to use their favorite technology, such as D3[2], HTML Canvas, or WebGL. The server runtime of Caleydo Web is implemented in Python using the Flask[2] framework. First individual plugins provide access to data storage files in HDF[3] or CSV format, and databases including Neo4j[4]. We plan to integrate an R interface for more complex data processing operations.

6 Demo Application: StratomeX.js

StratomeX.js is a Caleydo Web-based reimplementation of Caleydo *StratomeX* [5], a cancer subtype visualization technique. Figure 2 shows a screenshot of the application with annotations indicating individual plugins of Caleydo Web, highlighting its reuseability. A demo version is available at `http://caleydo-web.herokuapp.com/stratomex.js`.

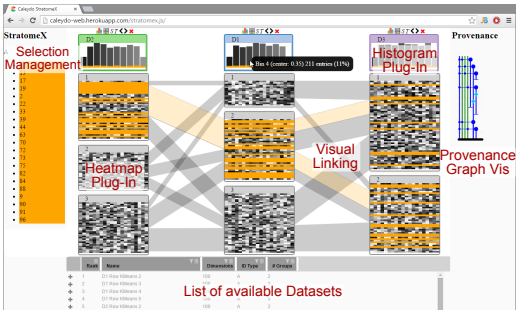

Figure 2: *Screenshot of StratomeX.js based on Caleydo Web. Individual components of Caleydo Web are annotated.*

Acknowledgements

This work was funded by the Austrian Research Promotion Agency (FFG) (840232), the Austrian Science Fund (FWF) (P27975-NBL and J 3437-N15), and the US National Institutes of Health (K99 HG007583, U01 CA198935).

References

[1] C. G. Begley and L. M. Ellis. Drug development: Raise standards for preclinical cancer research. *Nature*, 483(7391):531–533, Mar. 2012.

[2] M. Bostock, V. Ogievetsky, and J. Heer. D3: Data-Driven Documents. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '11)*, 17(12):2301–2309, 2011.

[3] J. G. et al. BioJS: an open source JavaScript framework for biological data visualization. *Bioinformatics*, 29(8):1103–1104, 2013.

[4] A. Lex, M. Streit, E. Kruijff, and D. Schmalstieg. Caleydo: Design and Evaluation of a Visual Analysis Framework for Gene Expression Data in its Biological Context. In *Proceeding of the IEEE Symposium on Pacific Visualization (PacificVis '10)*, pages 57–64. IEEE, 2010.

[5] A. Lex, M. Streit, H.-J. Schulz, C. Partl, D. Schmalstieg, P. J. Park, and N. Gehlenborg. StratomeX: Visual Analysis of Large-Scale Heterogeneous Genomics Data for Cancer Subtype Characterization. *Computer Graphics Forum (EuroVis '12)*, 31(3):1175–1184, 2012.

[6] T. Mühlbacher, H. Piringer, S. Gratzl, M. Sedlmair, and M. Streit. Opening the Black Box: Strategies for Increased User Involvement in Existing Algorithm Implementations. *IEEE Transactions on Visualization and Computer Graphics (VAST '14)*, 20(12):1643–1652, 2014.

[7] M. Nielsen. A guide to the day of big data. *Nature*, 462(7274):722–723, 2009.

[8] M. Streit, A. Lex, S. Gratzl, C. Partl, D. Schmalstieg, H. Pfister, P. J. Park, and N. Gehlenborg. Guided visual exploration of genomic stratifications in cancer. *Nature Methods*, 11(9):884–885, 2014.

[2] `http://flask.pocoo.org/`

[3] `http://www.hdfgroup.org/`

[4] `http://neo4j.com/`