From Microbes to Medical Cohorts: Visualizing Multi-Omics Data on Different Scales

Dissertation

der Mathematisch-Naturwissenschaftlichen Fakultät

der Eberhard Karls Universität Tübingen

zur Erlangung des Grades eines

Doktors der Naturwissenschaften

(Dr. rer. nat.)

vorgelegt von

Theresa Harbig

aus Bad Friedrichshall

Tübingen

2025

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Eberhard Karls Universität Tübingen.

Tag der mündlichen Qualifikation:	29.04.2025
Dekan:	Prof. Dr. Thilo Stehle
1. Berichterstatter/-in:	Prof. Dr. Kay Nieselt
2. Berichterstatter/-in:	Prof. Dr. Daniel Huson

"*Wege entstehen dadurch dass man sie geht.*"

Franz Kafka

Abstract

Biological data is analyzed in many domains, most importantly, in medical research to improve human well-being. The data can be used to study diseases and to find cures such as novel antibiotics when fighting antibiotic resistance. For this, researchers study how antibiotics act on bacteria and how bacteria themselves produce antibiotics to find potential candidate drugs. High-throughput methods, including second-generation sequencing or mass spectrometry can produce data providing a complete picture of an organism's genome, transcriptome, proteome, and metabolome. Each corresponding field is called an "omics" field and the combination is known as multi-omics.

Multi-omics data can be seen as a multi-layer network of genes, transcripts, proteins, and metabolites with interactions within and between omics layers. The shape of data is variable. Data sets may include many omics layers or only a subset, consist of few samples or entire cohorts, and can be of a single well-annotated species or a non-model organism. This complex data requires advanced analysis and visualization methods for its interpretation tailored to the underlying data and the biological questions. Visualization can help communicate analysis results to domain researchers and leverage the capacity of human brains for pattern recognition and integration of background knowledge. This dissertation presents exploratory visualization approaches for multi-omics data. Depending on the research question, the approaches are focused on different multi-omics levels and types of data sets. Furthermore, they apply several different integration methods, including knowledge-based integration, data-driven integration, and composite networks.

Knowledge-based integration combines omics data with known structures. With an approach for visualizing Biosynthetic Gene Clusters, this dissertation exemplifies how data can be integrated by mapping it to the coordinates of a genome and applying prediction algorithms to find genomic features of interest. In another approach

knowledge is integrated using universal vocabulary in the form of Gene Ontology terms to summarize the functional changes when comparing experiments or omics layers.

Data-driven integration integrates omics data without previous knowledge of annotations. For this, two visualization approaches presented help explore algorithmically or manually created groups or clusters of omics data. This is done at different scales, such as experiments with multiple conditions of a single organism or cohort data with many individuals. In an approach aimed at multi-condition experiments genes, transcripts, and proteins are clustered algorithmically to compare similar behavior across conditions. Similarly, in another approach cohorts of patients are grouped by multi-omics data or metadata and the temporal evolution and similarity of patient subgroups can be studied.

Finally, composite networks can combine omics data from different sources. This includes knowledge-based integration, for example by creating networks based on known protein-protein associations, or data-driven integration when building multi-omics correlation networks. This type of integration opens up the whole world of analysis and visualization using network approaches. Composite networks often show the small world property, meaning they have a small average path length and a high clustering coefficient. The network visualization presented in this dissertation leverages this property to reduce network size and display entire multi-omics networks.

Based on the presented approaches the different strategies for multi-omics integration and visualization are highlighted. Furthermore, their implementation is described and their usage is exemplified. Finally, this dissertation discusses how the approaches can be extended and made available to larger audiences.

Deutsche Zusammenfassung

Biologische Daten werden in vielen Bereichen analysiert, vor allem zur Verbesserung des menschlichen Wohlergehens bei der Erforschung von Krankheiten, um Heilmittel zu finden, wie neuartige Antibiotika zur Bekämpfung der Antibiotikaresistenz. Um neuartike Antibiotika oder Wirkmechanismen zu entdecken, untersuchen Forscher, wie Antibiotika von Bakterien produziert werden und wie sie auf Bakterien wirken. Hierzu werden Hochdurchsatzmethoden wie *Second Generation Sequencing* oder die Massenspektrometrie genutzt um Daten zu erzeugen, die ein vollständiges Bild des Genoms, Transkriptoms, Proteoms und Metaboloms eines Organismus vermitteln. Das entprechende Forschungsfeld wird als "omics" und die Kombination wird als "multi-omics" bezeichnet.

Multi-omics-Daten können als ein mehrschichtiges Netzwerk von Genen, Transkripten, Proteinen und Metaboliten mit Interaktionen innerhalb und zwischen omics-Schichten betrachtet werden. Die Form der Daten kann variabel sein. Datensätze können mehrere omics-Schichten oder nur eine Teilmenge umfassen, aus wenigen Proben oder ganzen Kohorten bestehen und von einer einzigen gut annotierten Spezies oder einem Nicht-Modellorganismus stammen. Die Interpretation dieser komplexen Daten erfordert spezialisierte Analyse- und Visualisierungsmethoden, die auf die zugrunde liegenden Daten und biologischen Fragen zugeschnitten sind. Visualisierung kann dazu beitragen, die Analyseergebnisse an die Fachwissenschaftler zu vermitteln und die Fähigkeit des menschlichen Gehirns zur Mustererkennung und Integration von Hintergrundwissen zu nutzen. In dieser Dissertation werden explorative Visualisierungsansätze für Multi-omics-Daten vorgestellt. Abhängig von der Forschungsfrage sind die Ansätze auf verschiedene Multi-omics-Ebenen und Arten von Datensätzen ausgerichtet. Außerdem werden verschiedene Integrationsmethoden angewandt, darunter wissensbasierte Integration, datengesteuerte Integration und zusammengesetzte Netzwerke.

Bei der wissensbasierten Integration werden Omics-Daten mit bekannten Strukturen kombiniert. Diese Dissertation zeigt beispielhaft, wie Daten integriert werden können, indem sie auf die Koordinaten eines Genoms abgebildet werden und Vorhersagealgorithmen angewendet werden, um genomische Merkmale von Interesse zu finden. Darüber hinaus wird Wissen durch universelles Vokabular integriert, indem die *Gene Ontology* verwendet wird um funktionelle Veränderungen beim Vergleich von Experimenten oder Omics-Ebenen zusammenzufassen.

Die datengesteuerte Integration kombiniert Omics-Daten ohne vorherige Kenntnis der Annotationen. Hier können Visualisierungstools helfen, algorithmisch oder manuell erstellte Gruppen oder Cluster von Omics-Daten zu untersuchen. Dies ist auf verschiedenen Ebenen möglich, z. B. bei der Analyse kontrollierter Experimente unter mehreren Bedingungen oder bei der Analyse von Kohortendaten vieler Individuen. Bei einem Ansatz, der auf Experimente mit mehreren Bedingungen abzielt, werden Gene, Transkripte und Proteine algorithmisch geclustert, um ähnliches Verhalten unter verschiedenen Bedingungen zu vergleichen. In ähnlicher Weise werden bei einem anderen Ansatz Patientenkohorten anhand von Multi-omics-Daten oder Metadaten gruppiert, um Patientengruppen zu vergleichen und ihre zeitliche Entwicklung zu untersuchen.

Darüber hinaus können zusammengesetzte Netzwerke Omics-Daten aus verschiedenen Quellen kombinieren. Dazu gehört die wissensbasierte Integration, z. B. durch die Erstellung von Netzwerken auf der Grundlage bekannter Protein-Protein-Assoziationen, oder die datengesteuerte Integration beim Aufbau von Multi-omics-Korrelationsnetzwerken. Diese Art der Integration ermöglicht die Analyse und Visualisierung mit etablierten Netzwerkansätzen. Zusammengesetzte Netzwerke weisen häufig die “kleine Welt” Eigenschaft auf, d. h. sie haben eine geringe durchschnittliche Pfadlänge und einen hohen Clustering-Koeffizienten. Die in dieser Dissertation vorgestellte Netzwerkvisualisierung macht sich diese Eigenschaft zunutze, um die Netzwerkgröße zu reduzieren und Multi-omics-Netzwerke in ihrer Gesamtheit darzustellen.

Basierend auf den vorgestellten Ansätzen werden die verschiedenen Strategien zur Multi-Omics-Integration und Visualisierung aufgezeigt. Darüber hinaus wird ihre Implementierung beschrieben und ihre Verwendung veranschaulicht. Abschließend wird disskutiert, wie die Ansätze erweitert und einem größeren Publikum zugänglich gemacht werden können.

Acknowledgements

This endeavor would not have been possible without my supervisors Prof. Dr. Kay Nieselt and Prof. Dr. Michael Krone. I want to thank you for your valuable expertise and moral support. Furthermore, I would like to thank Prof. Dr. Daniel Huson for marking my thesis, and the entire defense committee for facilitating my defense and for the fruitful discussions. Special thanks go to the TRR for funding my research and sponsoring my research trip to the US, and to the HIDIVE group led by Prof. Dr. Nils Gehlenborg for hosting me.

Moreover, I want to thank the collaborators from the TRR and all other collaborators for the interesting insights into their research questions and for joint publications. Specifically, I want to thank Mathias Witte Paz, Julian Fratte, Sabrina Nusrat, and Nicolas Brich for their valuable contributions to the publications featured in this thesis. You made developing new approaches much more fun.

Furthermore, I want to express my gratitude to all my peers from the Integrative Transcriptomics group, and the other groups with which I had lunch and coffee breaks and after-work hangouts. Special thanks to Susanne Zabel, who always gave me moral support, even when we could not physically share our office during the COVID-19 pandemic.

I would like to express my deepest gratitude to my family and friends for their support not only during my PhD, but also during my entire journey from the beginning of my bachelor's to my defense presentation, including my brothers and their families, my partner Nicolas, my school friends, and my parents.

Contents

List of figures

List of Tables

Contributions

Thesis Contributions

Chapter 4:	Visualizing Gene Function			
Author Position & Author	Scientific Ideas	Data Generation & Implementation	Analysis & Interpretation	Paper Writing
1. Harbig, Theresa	85 %	100 %	60 %	70 %
2. Witte Paz, Mathias	10 %	0 %	30 %	20 %
3. Nieselt, Kay	5 %	0 %	10 %	10 %
Title of Paper:	GO-Compass: Visual Navigation of Multiple Lists of GO Terms			
Status:	published			

Chapter 5:	Visualizing Cluster Patterns			
Author Position & Author	Scientific Ideas	Data Generation & Implementation	Analysis & Interpretation	Paper Writing
1. Harbig, Theresa	45 %	70 %	45 %	65 %
2. Fratte, Julian	40 %	30 %	45 %	20 %
3. Krone, Michael	5 %	0 %	5 %	5 %
4. Nieselt, Kay	10 %	0 %	5 %	10 %
Title of Paper:	OmicsTIDE: Interactive Exploration of Trends in Multi-Omics Data			
Status:	published			

Chapter 6:	Visualizing Cohorts			
Author Position & Author	Scientific Ideas	Data Generation & Implementation	Analysis & Interpretation	Paper Writing
1. Harbig, Theresa	30 %	60 %	35 %	40 %
2. Nusrat, Sabrina	22.5 %	40 %	30 %	30 %
3. Tali, Mazor	12.5 %	0 %	10 %	10 %
4. Wang, Quianwen	2.5 %	0 %	5 %	5 %
5. Thomson, Alexander	5 %	0 %	2.5 %	0 %
6. Bitter, Hans	5 %	0 %	2.5 %	0 %
7. Cerami, Ethan	10 %	0 %	5 %	5 %
8. Gehlenborg, Nils	12.5 %	0 %	10 %	10 %
Title of Paper:	OncoThreads: visualization of large-scale longitudinal cancer molecular data			
Status:	published			

Chapter 7:	Visualizing Networks			
Author Position & Author	Scientific Ideas	Data Generation & Implementation	Analysis & Interpretation	Paper Writing
1. Brich, Nicolas	35 %	45 %	30 %	25 %
2. Harbig, Theresa	35 %	35 %	30 %	35 %
3. Witte Paz, Mathias	15 %	20 %	25 %	20 %
4. Nieselt, Kay	7.5 %	0 %	5 %	5 %
5. Krone, Michael	7.5 %	0 %	10 %	15 %
Title of Paper:	ProtEGOnist: Visual Analysis of Interactions in Small World Networks sing Ego-graphs			
Status:	published			
Note:	The first two authors contributed equally and should be regarded as joint first authors. This Paper will also be included as a chapter in the dissertation of Nicolas Brich.			

Other Contributions

S. Wilcken, P.-H. Koutsandrea, T. Bakker, A. Kulik, T. Orthwein, M. Franz-Wachtel, T. Harbig, K. K. Nieselt, K. Forchhammer, H. Brötz-Oesterhelt, B. Macek, S. Mordhorst, L. Kaysser, and B. Gust, "The TetR-like regulator Sco4385 and crp-like regulator Sco3571 modulate heterologous production of antibiotics in streptomyces coelicolor M512," en, *Appl. Environ. Microbiol.*, vol. 91, no. 5, e0231524, May 2025, ISSN: 0099-2240,1098-5336. DOI: 10.1128/aem.02315-24

L. Schulze, J. Moessner, S. Krauss, T. Harbig, K. Nieselt, B. Krismer, and A. Peschel, "Genetic modification of intractable staphylococcal clones by heat-shock facilitated phage transduction," *bioRxiv*, Apr. 2025. DOI: 10.1101/2025.04.04.647181

N. Gericke, D. Beqaj, T. Kronenberger, A. Kulik, A. Gavriilidou, M. Franz-Wachtel, U. Schoppmeier, T. Harbig, J. Rapp, I. Grin, N. Ziemert, H. Link, K. Nieselt, B. Macek, W. Wohlleben, E. Stegmann, and S. Wagner, "Unveiling the substrate specificity of the ABC transporter tba and its role in glycopeptide biosynthesis," en, *iScience*, vol. 28, no. 4, p. 112 135, Apr. 2025, ISSN: 2589-0042. DOI: 10.1016/j.isci.2025.112135

S. Hackl, C. Jachmann, M. Witte Paz, T. A. Harbig, L. Martens, and K. Nieselt, "PTMVision: An interactive visualization webserver for post-translational modifications of proteins," en, *J. Proteome Res.*, Jan. 2025, ISSN: 1535-3893,1535-3907. DOI: 10.1021/acs.jproteome.4c00679

A. Hoffmann, U. Steffens, B. Maček, M. Franz-Wachtel, K. Nieselt, T. A. Harbig, K. Scherlach, C. Hertweck, H.-G. Sahl, and G. Bierbaum, "The unusual mode of action of the polyketide glycoside antibiotic cervimycin C," en, *mSphere*, vol. 9, no. 5, e0076423, May 2024, ISSN: 2379-5042. DOI: 10.1128/msphere.00764-23

M. Bianchi, M. Winterhalter, T. A. Harbig, D. Hörömpöli, I. Ghai, K. Nieselt, H. Brötz-Oesterhelt, C. Mayer, and M. Borisova-Mayer, "Fosfomycin uptake in escherichia coli is mediated by the outer-membrane porins OmpF, OmpC, and LamB," en, *ACS Infect Dis*,

vol. 10, no. 1, pp. 127–137, Jan. 2024, ISSN: 2373-8227. DOI: 10.1021/acsinfecdis.3c00367

R. Shukla, A. J. Peoples, K. C. Ludwig, S. Maity, M. G. N. Derks, S. De Benedetti, A. M. Krueger, B. J. A. Vermeulen, T. Harbig, F. Lavore, R. Kumar, R. V. Honorato, F. Grein, K. Nieselt, Y. Liu, A. M. J. J. Bonvin, M. Baldus, U. Kubitscheck, E. Breukink, C. Achorn, A. Nitti, C. J. Schwalen, A. L. Spoering, L. L. Ling, D. Hughes, M. Lelli, W. H. Roos, K. Lewis, T. Schneider, and M. Weingarth, "An antibiotic from an uncultured bacterium binds to an immutable target," en, *Cell*, vol. 186, no. 19, 4059–4073.e27, Sep. 2023, ISSN: 0092-8674,1097-4172. DOI: 10.1016/j.cell.2023.07.038

M. D. Mungan, T. A. Harbig, N. H. Perez, S. Edenhart, E. Stegmann, K. Nieselt, and N. Ziemert, "Secondary metabolite transcriptomic pipeline (SeMa-trap), an expression-based exploration tool for increased secondary metabolite production in bacteria," en, *Nucleic Acids Res.*, vol. 50, no. W1, W682–9, May 2022, ISSN: 0305-1048,1362-4962. DOI: 10.1093/nar/gkac371

S. Krauss, T. A. Harbig, J. Rapp, T. Schaefle, M. Franz-Wachtel, L. Reetz, A. M. A. Elsherbini, B. Macek, S. Grond, H. Link, K. Nieselt, B. Krismer, A. Peschel, and S. Heilbronner, "Horizontal transfer of bacteriocin biosynthesis genes requires metabolic adaptation to improve compound production and cellular fitness," en, *Microbiol Spectr*, e0317622, Dec. 2022, ISSN: 2165-0497. DOI: 10.1128/spectrum.03176-22

S. T. Hackl, T. A. Harbig, and K. Nieselt, "Technical report on best practices for hybrid and long read de novo assembly of bacterial genomes utilizing illumina and oxford nanopore technologies reads," en, *bioRxiv*, p. 2022.10.25.513682, Oct. 2022. DOI: 10.1101/2022.10.25.513682

A. Dietrich, U. Steffens, M. Gajdiss, A.-L. Boschert, J. K. Dröge, C. Szekat, P. Sass, I. T. Malik, J. Bornikoel, L. Reinke, B. Maček, M. Franz-Wachtel, K. Nieselt, T. Harbig, K. Scherlach, H. Brötz-Oesterhelt, C. Hertweck, H.-G. Sahl, and G. Bierbaum, "Cervimycin-resistant staphylococcus aureus strains display vancomycin-intermediate resistant phenotypes," en, *Microbiol.*

Spectr., vol. 10, no. 5, e0256722, Oct. 2022, ISSN: 2165-0497. DOI: 10.1128/spectrum.02567-22

Q. Wang, T. Mazor, T. A. Harbig, E. Cerami, and N. Gehlenborg, "ThreadStates: State-based visual analysis of disease progression," en, *IEEE Trans. Vis. Comput. Graph.*, vol. 28, no. 1, pp. 238–247, Jan. 2022, ISSN: 1077-2626,1941-0506. DOI: 10.1109/TVCG.2021.3114840

M. Witte Paz, T. A. Harbig, and K. Nieselt, "Evidente—a visual analytics tool for data enrichment in SNP-based phylogenetic trees," en, *Bioinform. Adv.*, vol. 2, no. 1, Jan. 2022, ISSN: 2635-0041. DOI: 10.1093/bioadv/vbac075

S. Beier, A. Gorska, P. Grupp, T. A. Harbig, I. Flade, and D. H. Huson, "Bioinformatics support for the tubiom community gut microbiome project," en, PeerJ Preprints, Tech. Rep. e2382v1, Aug. 2016. DOI: 10.7287/peerj.preprints.2382v1

Chapter 1

Introduction

> *If a man has his eyes bound, you can encourage him as much as you like to stare through the bandage, but he'll never see anything. He'll be able to see only when the bandage is removed.*
>
> —Franz Kafka, *The Castle*

Data visualization is applied in almost all domains from social science, to engineering, to biology. Visualizations communicate data and produce insights by leveraging the visual processing capacity of the human brain. As Kafka would put it, staring at data tables or numbers and expecting to gain insight is like staring through a bandage and expecting to see. With visualization, the bandage can be lifted and the data and its properties can be understood. This is illustrated by the well-known example called Anscombe's quartet (Figure 1.1) [15]. Four data sets seem almost identical concerning statistical parameters including mean, variance, and correlation. Only when applying visualization, we can see that they are very different.

Visualization ranges from simple information visualization communicating election results to exploratory visualizations of bio-medical data. Exploring data using visualizations before or instead of statistical analysis is called *exploratory data analysis* and was first introduced by John Tukey in 1978 [16]. It is essential for data-driven projects, where huge amounts of data are produced, but the research question is not easily translated into a hypothesis for traditional hypothesis testing.

Despite advancements in computational techniques, such as artificial intelligence which can automate large parts of the analysis process, human involvement remains critical. Computational approaches do not replace the human in the loop but rather change

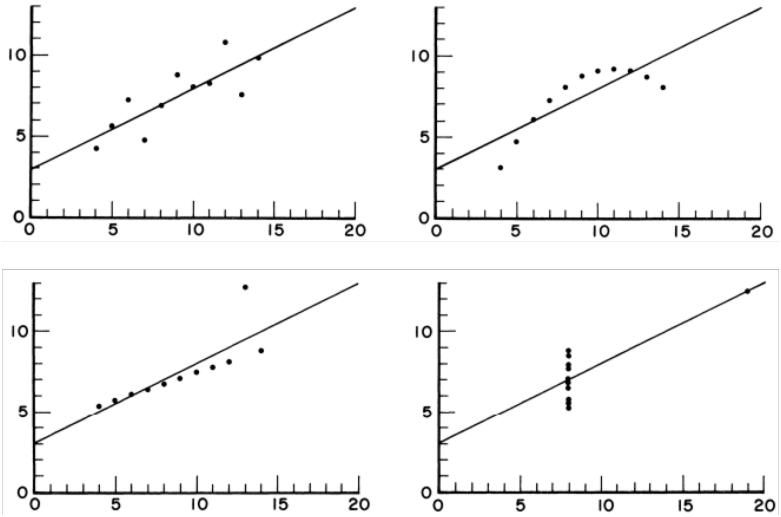


Figure 1.1: Ansombes Quartet [15]. Scatterplots of four data sets of x-y pairs. The data sets are almost identical concerning common statistical parameters ($\overline{x} = 9$,$\overline{y} = 7.5$, $s_x^2 = 11$, $s_y^2 = 4.125$, $r_{xy} = 0.816$). However, the visualizations make the differences of the data salient.
Source: Graphs in Statistical Analysis, F.J. Anscombe, The American Statistician, ©1973, reprinted by permission of Informa UK Limited, trading as Taylor & Taylor & Francis Group, http://www.tandfonline.com

at which point humans add their skills to an analysis. For this, the power of visualization is not only to communicate and explore but also to increase a researcher's confidence in the analysis process. After all, humans judge the results of algorithms and decide on the next steps. This highlights that when designing a visualization, the audience is important, and visualizations are to be interpreted with context. For this thesis, visualizations were developed with domain experts to understand how data can be communicated truthfully and efficiently.

1.0.1 Motivation

Much of the work presented in this dissertation was developed with domain experts from the transregional collaborative research center TRR 261 "Cellular Mechanisms of Antibiotic Action and Production" (acronym "ANTIBIOTIC CellMAP"). The center has studied how antibiotics are produced and how they act on bacteria in multiple research projects. Close collaboration in data analysis projects with domain experts led to a deeper understanding of the data and the general requirements for visualization tools.

The research projects aim to investigate antibiotic action and production mechanisms on several different levels, leading to complex multi-dimensional data (Figure 1.2a). For this, the differences between genomes are studied, for example how genomes of bacterial strains resistant to an antibiotic differ from sensitive strains. The corresponding field is called genomics. Moreover, the conditions causing the responsible genes to become active are explored in the fields of transcriptomics and proteomics. Finally, in metabolomics, the compounds produced that act as antibiotics and their intermediates are studied. The combination of genomics, transcriptomics, proteomics, and metabolomics data can be seen as a multi-layer network, where each field is called an "omics layer" (Figure 1.2b). Data exists (experimental or found in databases) that links entities within an omics layer, such as protein-protein interactions, or between omics layers, such as pathway data.

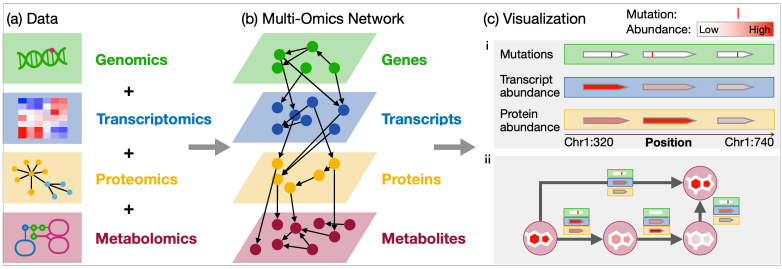


Figure 1.2: (a) Multi-Omics data consists of multiple layers, such as genomics, transcriptomics, proteomics, and metabolomics. Within the layers, different entities are studied, which are genes, transcripts, proteins, and metabolites. (b) The data can be seen as a multi-layer network, since the entities are connected within and between the layers. For example, genes belonging to the same family can be linked within the genomic layer, but they are also linked to their corresponding transcripts, proteins, and the involved metabolites. (c) Data can be visualized by (i) mapping genomic, transcriptomic, and proteomic data on the corresponding genes in a genome-coordinate visualization or (ii) mapping all omics layers to a pathway network.

The data analysis projects in the TRR usually involved well-defined research questions and thus testable hypotheses. They involved analyzing transcriptomics data regarding differential expression and gene function, genomic analysis concerning the detection of *Single Nucleotide Polymorphisms* (SNPs), and joint interpretation of genomics, transcriptomics, and proteomics data [6], [7], [9], [11]. Many of these analyses produce tables, such as tables of differential expression analysis, tables of SNPs, or tables of statistically over-represented functional terms. For statistical analysis, R packages, such as `DeSeq2` [17], are often used, including the creation plots like gene expression heatmaps, dendrograms, or Venn diagrams. While these conventional methods can help answer the most immediate question, such as evaluating mutations within or the expression of genes of interest, the amount of data produced calls for exploratory methods. The methods should facilitate a holistic analysis of multi-omics data and the discovery of unexpected results.

1.0.2 Exploratory Multi-Omics Visualization

For the most basic visualization, multi-omics data can be displayed by mapping them to structures inherent to the data. For example, all gene-based data, such as genomic, transcriptomic, and proteomic data can be mapped to genes in genome coordinate visualizations, such as genome browsers (Figure 1.2c,i). The genome browser `IGV` is an approach for interactive genome visualization of sequencing data [18], which has various applications for multi-omics data. It can be applied as a visual quality control but also visualize SNPs and map numerical values (gene expression or protein abundance) to genes. Furthermore, metabolic pathways can be used for visualization, which show how genes catalyze reactions creating metabolites (Figure 1.2c,ii). Tools like `Vanted` [19] or `Pathview Web` [20] can be used to combine the individual results in one pathway view for a pathway of interest. Metabolic pathways are a type of network, which is a common structure for multi-omics data, including gene co-expression networks, or multi-omics networks. Yet, genomic-coordinate visualizations, metabolic pathways, and other network visualizations are suited for a small number of genes or nodes. They are only useful if the genes or pathways of interest are known, but become too large to interpret if many genes or nodes are displayed.

Multi-omics data sets can be very large as they often contain the entirety of genes, transcripts, proteins, and/or metabolites within one or many samples. Furthermore, multi-omics data sets are not only produced on a per-experiment basis, but databases exist with large amounts of data, adding background knowledge to the analysis. Databases can contain data on a single omics level, for example, protein structures, but also contain multi-omics data, such as protein-protein associations created using multi-omics data. Moreover, the shape of the multi-omics data sets is variable. Not all omics layers are always available when analyzing experimental data. Due to the cost of omics analyses, researchers carefully decide which omics layer is worth analyzing. Genomics and transcriptomics analysis are comparatively inexpensive as they apply next-generation sequencing techniques. In contrast, proteomics and metabolomics are more expensive due to the application of mass spectrometry methods. Moreover, not only the available layers can differ, but

also the number of samples and replicates. For the TRR, multi-omics data sets stem from controlled experiments with few conditions (usually less than 10) and few replicates (mostly not more than three). In contrast, cancer research deals with large cohorts of patients, each with a varying number of samples and replicates. In conclusion, multi-omics data is large, heterogeneous, and variable in shape, which represents a major challenge for analysis.

Furthermore, the specific research goals are critical. Much research is gene-based, for example, when studying human cancer or operon structures in prokaryotes. Here, the challenge is making potential genes of interest salient to the researcher and visualize them in context with other multi-omics data. At other times, transcriptomics and proteomics data are studied, for example, to gain insight into the molecular mechanisms of gene expression on a genome-wide level. Transcriptomics and proteomics data can easily be connected using shared identifiers, which can be leveraged for integration. Yet, the connection to metabolites or comparing different species where identifiers do not match is not as simple. Species-agnostic vocabulary, such as Gene Ontology terms can be used to translate the function of molecular entities into a universal language. Alternatively, a network modeling the connections between genes, transcripts, proteins, and metabolites can be the basis of analysis. However, multi-omics research is not always focused on molecular entities, but also on large cohorts of patients. Here, the goal is to find associations between disease phenotypes and multi-omics data. Similar to gene-based research, a main challenge is making multi-omics features of interest salient.

To address these research goals, the data must be transformed into a more interpretable shape with multi-omics data integration methods. This includes, for example, joint clustering of omics layers [21], the creation of multi-omics networks [22], or overrepresentation tests and gene set enrichment analysis [23]. After integration, a multitude of analysis and visualization methods can be applied. For example, when a multi-omics network is created, the whole repertoire of network methods can be applied for analysis and visualization. Yet, visualization approaches for multi-omics data should be accessible to users without advanced computational skills, which is important

when applying and explaining statistical computations, clustering, or advanced machine-learning techniques in a visualization.

The work presented in this dissertation provides visualization approaches that make multi-omics data interpretable for domain experts. As reflected in the title of this dissertation, data is visualized on different scales, from gene-centric analysis in microbes to analyzing entire single-omics layers, multiple omics layers, and even analyzing *many* individuals in cohorts. Moreover, this dissertation aims to showcase how approaches can be designed systematically by abstracting the underlying data and the research space.

1.0.3 Contributions

This dissertation first provides an overview of the data and visualization techniques needed to design and implement effective data visualization approaches (chapter 2). Then, the developed approaches are described and discussed (chapters 3-7) in-depth followed by a joint discussion (chapter 8).

In chapter 3 visualization techniques for genomic visualizations are presented. The chapter highlights how knowledge of the visualization research space obtained by contributing to a *State of the Art Report* (STAR) [24] can be combined with domain expertise for systematic visualization development. The first presented approach, `SeMa-Trap`, is a visualization for the results of a pipeline detecting *Biosynthetic Gene Clusters* (BGCs) producing *secondary metabolites* in prokaryotes and quantifying their expression [8]. The visualization is based on an extensive computational pipeline, which only requires raw transcriptomics data, and is easy to apply by domain scientists. As a more universal, pipeline-independent approach `Gosling-Meta` is presented, an extension of the genome-coordinate visualization grammar `Gosling` [25]. `Gosling-Meta` extends this with meta-data visualizations, thus contributing to the quick development of genome-coordinate-based visualization approaches.

Often, not only single entities such as BGCs are of interest, but the goal is gaining an overview of one or even multiple omics layers. Therefore, frequently statistical analyses are performed to find genes or metabolites with significantly altered expression between

two conditions. The approach GO-Compass presented in chapter 4 provides a comparative visualization for the functional composition of multiple lists of genes [26]. GO-Compass was inspired by the collaborations within the TRR, where a common issue in analyzing lists of differentially expressed genes was the high number of genes. Typically, genes are categorized into pathways or functional terms for interpretation. The Gene Ontology provides such vocabulary in the shape of GO terms. Overrepresentation or enrichment tests are applied to find overrepresented terms, resulting in lists of GO terms that can still be too long and redundant for manual interpretation. GO-Compass comparatively visualizes such lists and reduces their redundancy. With this novel approach inter-species comparison and combinations of any omics layer translatable into GO terms are now possible.

While the previously described approaches integrate multi-omics data based on existing knowledge and metadata, other integration approaches are based on the experimental data alone. One of the most common combinations of omics data is transcriptomics and proteomics data. This combination is especially worthwhile in a multi-omics analysis as protein-coding transcripts and proteins often have a 1 to 1 relationship in particular in prokaryotic organisms where splicing does not occur. OmicsTIDE exploits this feature and combines the two omics layer by their shared identifier [27] (chapter 5). It visualizes transcript and protein abundance for a single organism across different conditions and performs joint multi-omics clustering. One of the central aims of OmicsTIDE is to find correlated subgroups of genes that show concordant patterns across two omics layers, which we identified as a common task in the TRR.

Multi-omics data can be visualized on larger scales, where entire cohorts of individuals are visualized. OncoThreads presented in chapter 6 maps multi-omics data to patient samples [28]. The approach visualizes the development of a cohort over time according to multiple selected multi-omics data or metadata features, such as the expression of specific genes. Using computationally aided visualization methods, features of interest can be identified and used for visualization.

As described above, composite networks are a common technique for integrating multi-omics data. They can be studied using `ProtEGOnist` [29], an approach for visualizing small-world networks presented in chapter 7. The approach was initially developed to contribute to the Bio+MedVis Challenge 2023 to visualize protein-protein interaction networks and was declared the winner. As input data, a network of the `STRING` database was used, which includes proteins and their associations, which are based on data of several omics levels, such as gene co-expression (transcriptomics), genomic neighborhood (genomics), protein homology (proteomics), and text mining (any omics) [30]. After the challenge the approach has been extended to visualize any small-world network, such as multi-omics correlation networks, or—as demonstrated in the corresponding chapter—social networks. The approach is based on ego graphs and makes impactful nodes, such as proteins with many associations salient in the visualization.

In conclusion, this thesis contributes novel approaches for visualizing multi-omics data using different data integration methods and focusing on different multi-omics layers and scales. The methods are based on different data integration methods, such as mapping multi-omics data to genomes (`SeMa-Trap`) or universal vocabulary (`GO-Compass`), using the data for joint clustering (`OmicsTIDE`), mapping the data to patients (`OncoThreads`), or creating composite multi-omics networks (`ProtEGOnist`). While some approaches were developed with a specific layer as a focus, such as data mappable to genes in `SeMa-Trap` and `OmicsTIDE`, the others represent more generalizable approaches. The approaches are focused on different scales, ranging from single locations on genomes in `SeMa-Trap` to data of entire medical cohorts in `OncoThreads`.

Chapter 2

Background

Analyzing multi-omics data aims to gain insight into complex biological processes (Figure 2.1a). Figure 2.1b-d illustrates the steps required for performing multi-omics experiments. High-throughput methods are used to produce data, which is analyzed in computational analysis pipelines and integrated using multi-omics integration methods. The analysis results are explored and communicated using visualization to gain meaningful insights. Every step in this process influences the final result of the analysis. Thus, understanding the complexity of biological processes, data generation, and data analysis is crucial for interpreting results and developing data-exploration approaches for integrated multi-omics data. This chapter describes the biological processes, omics data generation and analysis, and multi-omics integration. Finally, the role of visualization and its importance for exploring and communicating complex (in particular biological) data on different scales is covered.

2.1 Multi-Omics Data

The role of the molecules studied must be understood for a mental model of multi-omics data. Every organism consists of proteins, fats, sugars, and complex metabolites. These molecules are created in biochemical reactions facilitated by specialized proteins called enzymes. The building plans for these molecules are stored in *Deoxyribonucleic acid* (DNA) in every cell. DNA consists of nucleotides, each composed of a sugar, a phosphate group, and one of four nucleobases: Adenine (A), Guanine (G), Cytosine (C), and Thymine (T) [31]. Chains of nucleotides form a DNA strand. Two DNA strands are wound up as a double helix and attached at the nucleobases. Due to their structure, each nucleobase typically pairs

with a complementary base (A and T as well as C and G) forming so-called Watson-Crick pairs.

The Central Dogma of Molecular Biology states that the DNA is transcribed into single-stranded messenger RNA (mRNA) in a process called transcription, The mRNA is then translated into proteins in a process called translation [32] (Figure 2.1a). Not the entire DNA is transcribed, but subsequences of the genome called *genes*. Each gene has a specific function, often encoding for a certain enzyme. The process of transcription and translation is subsumed as gene expression. Gene expression is tightly regulated, where transcription controls the amount of mRNA produced from a specific gene, and translation subsequently controls the number of proteins made from the mRNA copies. Depending on external factors, different genes are transcribed and translated to trigger different biochemical reactions, adapting an organism to its environment.

Each step in this process and the resulting molecules are studied to analyze the characteristics of an organism. For example, the human genome can be analyzed to find characteristics determining human traits [33]. The corresponding field in molecular biology studying genomes is called *genomics*. Furthermore, in the fields *transcriptomics*, *proteomics*, and *metabolomics* transcripts, proteins, and metabolites are analyzed to explore the effect of stimuli on their expression, for example when analyzing which effect different types of antibiotics have on the organism.

Figure 2.1 shows an overview of data analysis of the different omics fields (Figure 2.1a-c) and their integration (Figure 2.1d). The high-throughput methods used for data acquisition differ for the omics layers and produce different types of data (Figure 2.1b), each processed with specialized bioinformatics methods (Figure 2.1c). Each processing method produces features that can be of interest for data analysis and integration, such as expression rates or genomic mutations. The following sections describe in detail how the layers are processed and how data is integrated using multi-omics integration methods (Figure 2.1d).

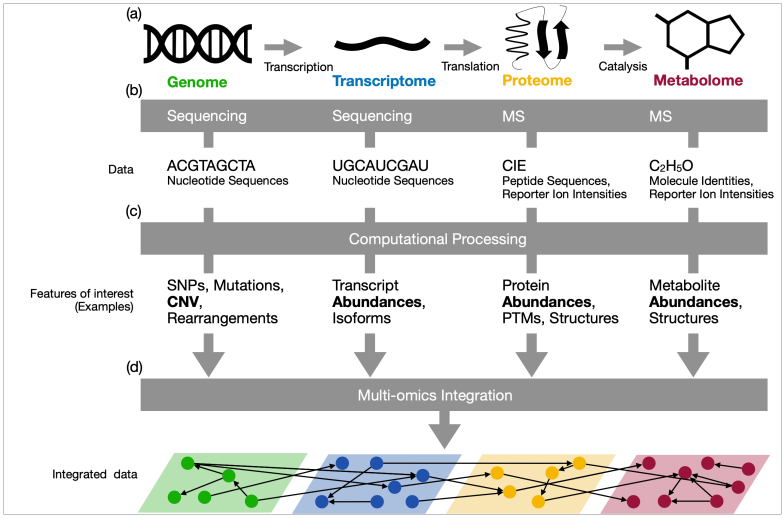


Figure 2.1: Steps of multi-omics data analysis. Multi-omics data contains data from genomes, transcriptomes, proteomes, and metabolomes. Genes are transcribed into transcripts, and transcripts are translated into proteins. The proteins often act as enzymes in metabolic pathways catalyzing reactions that build metabolites. The high throughput methods for omics layers differ. Genomes and transcriptomes are sequenced, while proteomes and metabolomes are studied using *mass spectrometry* (MS). The results are nucleotide sequences for the genome and transcriptome, peptide sequences for the proteome, and molecule identities for the metabolome together with their abundances. The data is computationally processed using various bioinformatics methods to extract features of interest. For multi-omics integration, especially abundance data (highlighted in bold) is of interest. Multi-omics integration methods are applied to connect the different multi-omics layers

2.1.1 From Genes to Metabolites

In genomics, the impact of alterations or natural differences between genomes is studied. Therefore, it is crucial to understand how a genome and genes are structured to distinguish genes from non-coding regions. In transcriptomics and proteomics knowledge about the mechanisms determining the transcription and translation rates is the basis for analyses. Furthermore, for metabolomics, we need to understand how metabolites determine an organism's phenotype.

Genes have conserved subsequences across different organisms. The cell apparatus uses specific subsequences within and around genes in transcription and translation to identify genes and control gene expression. During transcription enzymes, called RNA polymerases, bind to specific subsequences of the DNA and create an RNA copy of the gene. The subsequence is located upstream of a gene and is called a *promoter*. The polymerase unwinds the DNA starting at the promoter and starts transcription at the so-called *transcription start site*. From there, it traverses the gene using base pairing to create a complementary RNA molecule.

Other subsequences are important for controlling which genes are transcribed and at which intensity. Transcription factors are proteins that regulate transcription by binding to *transcription factor binding sites*. They act by increasing or blocking the binding of polymerase to the DNA or assisting in unwinding the DNA, making it accessible for transcription.

Different types of RNA are created during transcription. The messenger RNA (mRNA) is the template for encoding proteins, while ribosomal RNA (rRNA), transfer RNA (tRNA), and regulatory RNAs control the translation process. For translation, multiple rRNAs together with ribosomal proteins form a ribosome which assembles around an mRNA molecule. tRNAs bring amino acids to the ribosome and are a component for the synthesis of proteins based on the genetic sequence.

The *genetic code* determines how a protein is built based on the genetic sequence [34]. Genes consist of triplets of bases called *codons* which encode for amino acids. The combination of bases in the codon determines the amino acid used for the protein. tRNAs are

attached with amino acids and bind to the mRNA using a sequence complementary to the codon called the *anti-codon*. Moreover, specific codons signal the start and end of transcription, called start and stop codons.

The resulting proteins can have structural and mechanical functions or act as enzymes in biochemical reactions producing metabolites. Metabolites and enzymes form complex metabolic pathways modeling the biochemical reactions of an organism. In conclusion, depending on the environment, an organism needs to adapt its phenotype and thus needs to activate different metabolic pathways. For example, when running, a person's cells require more energy, and thus glucose needs to be produced. Depending on the intensity and the physical state of the person different genes are transcribed and translated into proteins, which catalyze the biochemical reactions for glucose production.

2.1.2 Prokaryotes and Eukaryotes

Prokaryotes and eukaryotes differ on a cellular and genomic level. In general, the cellular structure, the processes of transcription and translation, and the structure of genes are more complex in eukaryotes. For multi-omics analyses, especially the differences affecting the DNA, RNA, and protein sequences are of interest.

While prokaryotic genomes are circular and (most) consist of a single sequence known as a chromosome, eukaryotic genomes consist of multiple linear chromosomes in one or multiple copies. Furthermore, the sequences used for the transcription and translation machinery are different for eukaryotes and prokaryotes. Typically, eukaryotic genes consist of non-coding parts called introns and coding parts called exons. Introns are spliced before translation. In the process known as alternative splicing, a set of exons is also spliced leading to a different protein product, called a protein isoform.

In general, prokaryotic genomes are densely packed with coding genes, while eukaryotes also include other genomic sequences, regulating gene expression or of unknown function. Moreover, prokaryotic genomes include *operons*, which are contiguous genes often involved in the same pathway, such as the biosynthesis of a compound in the case of *biosynthetic gene clusters*.

2.1.3 Genomics

Genomics aims at associating phenotypes with genomic information on different scales from studying different strains of bacteria, to cancer phenotypes in patient cohorts, to evolutionary history. Possible explanations for a phenotype can only be found by comparison. For example, the genomic origin of antibiotic resistance of a bacterial strain can be studied on a gene-level scale by sequencing and comparing it to a non-resistant strain [11], [35]. In contrast, *Genome-wide association studies* (GWAS) are used to associate genomic variants in large cohorts with a phenotypic trait. This is often used in cancer research to associate *Single Nucleotide Polymorphisms* (SNPs) with cancer types to find potential routes for treatment [36]–[38]. Finally, the genomes of different species can be compared to infer the evolutionary history of these species.

Genome Sequencing and Assembly

The sequencing method developed by Sanger enabled researchers to study and compare whole genomes computationally [39]. The method sequences DNA one base at a time in a linear sequential process and requires time-consuming manual preparation and electrophoresis steps. With next-generation sequencing, many genomes could be sequenced quickly and more cheaply. To this date, Illumina sequencing is widely applied [40]. In this technique, the DNA is fragmented and amplified using *polymorphase chain reaction* (PCR) [41]. Millions of fragments can be sequenced in parallel on a chip. The method uses *sequencing by synthesis*, which means that fluorescently labeled DNA polymerase enzymes add complementary bases to the single-stranded fragments. The resulting light signals are captured and used to deduct the sequence of a fragment called a *read*. The synthesis of the DNA strand becomes erroneous

after only a small number of base pairings. Therefore, Illumina reads only have a length of up to 300 base pairs. Third-generation sequencing techniques, such as nanopore sequencing, produce much longer reads (up to hundreds of kilobases) [42]. The technique works by traversing a strand of DNA through a membrane protein called a nanopore while measuring changes in electric current to identify bases. While nanopore reads are longer, they are often of worse quality than Illumina reads. Yet, advances in base calling and error correction methods continue to improve read quality [43].

Using computational alignment methods reads are assembled to obtain a contiguous sequence by finding overlaps between the reads [44], [45]. This process is complicated by repetitive structures in the genome, leading to ambiguous placement of reads. Depending on the length of the reads, the size of the genome, and the number of repeats assembly can be challenging.

Not every genome is assembled from scratch using *de-novo* assembly as described above, but a reference genome is used as a template [46]. Instead of merging the reads, they are mapped to the reference genome. Reference-based assembly is useful for finding local differences between the sequenced genome and the reference, such as SNPs. Global differences, such as rearrangements of longer sequences are hard to find using this approach as repeat structures sometimes cannot be resolved.

Assembled genomes are often deposited in databases and can be used for reference-based assembly [47], [48]. Yet, not all assembled genomes are of high quality. The quality and length of the underlying reads influence the resulting genome. Assembly using Illumina reads has difficulties in resolving repeat regions longer than the short read length. Assemblies with Nanopore reads do not suffer from this issue, as the reads are longer, but the lower quality can lead to uncertainty in the exact genomic sequence.

Steps in the assembly pipeline aim at alleviating these issues. The quality of the reads can be analyzed [49] and they can be preprocessed in a process called *trimming* [50], [51], where low-quality regions or unwanted sequence artifacts from the processing the lab are removed or whole low-quality reads can be filtered. Yet, gaps

in the assembly can occur if too many reads are of low quality or if the number of reads is too low concerning the length of the genome, leading to parts of the genome being insufficiently covered by reads. Moreover, the assembly algorithm can influence the result and has to be carefully chosen based on the type of organism, available resources, and the downstream analysis goal [52]. Improvements in read quality [43] and algorithms combining reads from different sequencing techniques [10], [53] lead to more available high-quality reference genomes.

Genome Annotation

Apart from locating the differences between genomes, the impacted genes must be identified. Therefore, it is essential to know the genes' positions, extent, and annotations. The genes are predicted using empirical or ab initio methods. In empirical methods, genes are predicted using real-world evidence, such as an existing characterized protein product. For identification of genes alignments with known genes or protein sequences are performed using, for example, BLAST [54]. In ab initio methods genes are predicted using the genomic sequence alone. For this, specific subsequences, such as promoters, transcription factors, as well as start and stop codons are searched for [55], [56]. While this is relatively simple for prokaryotes, the structures of genes in eukaryotes are not as well understood.

Furthermore, the genes have to be annotated functionally, which means that the type of the resulting protein and the biological processes it is involved in have to be identified. A gene's function can be determined experimentally using biological assays such as yeast-two-hybrid or microarrays. However, computational methods are frequently used to infer a gene's function from existing evidence. A protein's function is directly related to the protein structure and thus also to the amino acid sequence and the gene sequence. Similar to empirical gene prediction methods, the function of a homologous gene, i.e. a gene with a similar sequence in a different organism, can be used for functional annotation. Moreover, subsequences called motifs representing specific functional domains can be used and searched for in databases such as `PROSITE` [57].

Yet, the function of a protein is much more conserved in its structure than in the underlying amino acid sequence. Therefore, the 3D structure of a protein can also be used for annotation, especially for sequences with low sequence homology. First, the structure has to be predicted [58], [59], and then it can be screened against a database with existing structures such as PDB [60].

Moreover, often other information is incorporated into functional annotation. Gene expression data can be used to find co-expressed genes, which usually are involved in the same processes and thus functionally similar [61]. Moreover, in prokaryotes functionally similar genes are often located in spatial proximity, which can be used to predict function based on location [62], [63]. Finally, databases such as STRING contain networks of proteins where nodes represent proteins and links are based on protein associations, such as physical contact, co-expression, and literature co-occurrence [30]. These networks are used to characterize the function of a protein by analyzing similar proteins.

Classification systems exist, such as KEGG [64], Pfam [65], and the Gene Ontology [66], to provide a unified functional vocabulary for gene function. The Gene Ontology is a system across species that describes gene function according to the categories "biological process", "molecular function", and "cellular component" using Gene Ontology terms (GO terms). The GO terms are organized roughly hierarchically for each category, where the root is the parent category, such as "biological process", and the children are more specific processes. The higher the distance to the root, the more specific the functional annotation. Genes can be associated with multiple GO terms.

2.1.4 Transcriptomics

Transcriptomics aims to explore how the transcriptome is affected by external stimuli. For this, the abundance of transcripts is measured. Starting in the 1990s [67], microarrays were the go-to method for measuring transcript abundances in a high-throughput fashion. A microarray consists of spots, each attached with a high number of single-stranded DNA probe sequences. The unknown chemically

labeled target sequences corresponding to the transcripts are added, bind to the corresponding probe, and produce a light signal. The intensity of a spot's light signal corresponds to the transcript's abundance.

RNA Sequencing

With the advent of next-generation sequencing and the decrease in sequencing cost, RNA sequencing has replaced microarrays in many domains. For RNA-Sequencing RNA is converted to complementary DNA (cDNA) which is sequenced using the same methods used for DNA sequencing. In contrast to microarrays, RNA sequencing is not targeted and thus any transcript contained in the sample is sequenced. This can be an advantage as previously unknown transcripts can be detected. Moreover, the sequence itself can be analyzed. However, the untargeted approach requires removing RNA not of interest, such as rRNA, which otherwise dominates the signal leading to a bad coverage for other transcripts.

Like DNA reads, RNA-Seq reads can be assembled. The algorithms are adapted to the specific challenges of transcriptomic data such as genome isoforms and employ either a de-novo strategy [68] or use a reference [69]. Furthermore, transcripts are quantified by mapping RNA-Seq reads to a reference genome and counting the number of reads mapping to each gene or other regions of interest [70]. Similar to genomics, repeating subsequences can lead to issues in quantification and assembly. In quantification, multi-mapping reads lead to ambiguity since assigning their counts is unclear. Advanced counting approaches use the placement of uniquely mapping reads as a model for placing multi-mapping reads, assuming that multi-mapping reads follow the same patterns as neighboring unique reads [71].

Data Analysis

To analyze the quantitative data statistically, usually, multiple RNA samples are taken in the form of replicates and are compared to samples of a "neutral" condition used as a control. For example, the expression of bacteria treated with an antibiotic is compared

to untreated bacteria. The expression is normalized according to sequencing depth and transcript length and differential expression is calculated and tested using specific statistical tests [17], [72], [73]. Furthermore, methods for multiple testing correction are applied as thousands of transcripts are tested [74]. The resulting data consists of gene IDs associated with fold changes and a (corrected) p-value.

Oftentimes, the calculation of differentially expressed genes is sufficient to confirm or reject a research hypothesis, especially when researchers study the organism on a gene-level scale and know the function of specific genes that play a role in the mechanism studied. However, usually downstream analysis is performed to analyze the data on a genome-wide scale. Genes can be clustered into modules with similar expression [75], co-expression networks can be created [76], and gene function can be studied in-depth [77], [78]. Gene set enrichment analysis (GSEA) and overrepresentation tests (ORA) are methods to functionally characterize lists of genes, such as differentially expressed genes or clusters of genes [79], [80]. The methods use classification systems such as Gene Ontology, Pfam, or KEGG to find functions enriched or overrepresented in lists of genes. In principle, annotations of all genes are used as a background and statistical methods test which functions occur more often than expected by chance in the gene list of interest.

Most approaches presented in this dissertation use expression data as input. `SeMa-Trap` presented in chapter 3 visualizes gene expression of *Biosynthetic gene clusters* [8]. The approach `OmicsTIDE` presented in chapter 5 clusters genes into modules using their expression across multiple conditions and offers analysis of the modules using `PANTHER` [77] to test for overrepresentation of functions [27]. Among many other data types, `OncoThreads` visualizes gene expression data for patient cohorts [28]. `GO-Compass`, presented in chapter 4 is an approach for the downstream analysis of lists of enriched GO terms using visualization [26].

2.1.5 Proteomics

Similar to transcripts, the proteins contained in a sample can analyzed and quantified. Proteomics measures protein abundance directly, while transcriptomics measures transcripts, which are not always translated into proteins. In fact, the correlation of proteomics and transcriptomics data is low, possibly due to factors such as non-coding RNA and post-transcriptional and post-translational modification.

Yet, technical differences have to be taken into account. High throughput mass spectrometry measures the protein abundances within a sample. Proteins are broken down into peptides using enzymatic digestion. The mass spectrometer identifies peptide sequences and quantifies their intensity. Peptides are assigned to proteins using a database search and protein abundance is calculated using the peptide intensities. This can be as simple as summing up the intensities of unique peptides assigned to a protein. More advanced methods can be used with a special procession of ambiguous peptides and the calculation of protein abundance [81].

Especially low-abundance proteins are hard to detect using mass-spectrometry due to large differences in the magnitudes of protein abundances and effects of protein degradation. Furthermore, alternative splicing and conserved peptide sequences appearing in multiple proteins can make accurate identification hard. In principle, the downstream analyses of proteomics data resemble those of transcriptomics since both data types consist of genes associated with a quantitative value.

2.1.6 Metabolomics

Metabolites are small molecules produced in metabolic pathways. They are not one class of molecules but many, including lipids, carbohydrates, and amino acids. This makes measuring metabolites hard, as not one type of molecule has to be targeted, like in proteomics, but many different types. As for proteins, mass spectrometry is used for measurement of molecule abundances. Depending on the types of metabolites studied, separation techniques such as gas

chromatography or high-performance liquid chromatography are applied beforehand. The downstream analysis again resembles those of proteomics and transcriptomics.

Metabolites can be classified into primary metabolites and secondary metabolites. Primary metabolites are essential in the organism's survival, such as glucose which delivers energy. Secondary metabolites are not essential but might be beneficial, such as antibiotics produced by a bacterium in competition with another bacterium [82].

2.1.7 Multi-Omics

The described fields are known as "omics" fields. Other omics fields exist, such as epigenomics, or subfields of the described omics fields, such as the metabolomics subfields lipidomics and glycomics. Multiple omics fields can be analyzed simultaneously in a *multi-omics* analysis (Figure 2.1d). The underlying mental model is a complex multi-omics network, where each omics field represents a layer with nodes for genes, transcripts, proteins, and metabolites, and edges for interactions, such as co-expression or co-occurrence in a metabolic pathway. The network can have different scales. The number of omics layers present can differ for data sets and biological questions. Moreover, all nodes in the network (i.e. all genes, transcripts, proteins, and metabolites) can be of interest or only a subset of them. Furthermore, depending on the question the data can consist of few samples from controlled experiments, to data from entire databases, to data from patient cohorts.

The data can be integrated computationally in many different ways. Wörheide, Krumsiek, Kastenmüller, *et al.* divides the types of integration into *knowledge-based*, *data-driven*, and *composite networks* [83]. Knowledge-based methods associate multi-omics data with existing data from databases, such as the structure of pathways or existing networks. They include set-based methods, such as gene set enrichment methods and overrepresentation tests [79], [80], where overrepresented annotation terms are found for each omics layer and compared afterward. Data-driven methods work without existing annotations. They often apply machine learning techniques or

statistical methods to find relationships between omics layers and often also associate them with a phenotype. This can be as simple as applying a joint clustering to multiple omics layers but also includes predicting disease phenotypes using deep neural networks [84]. Composite networks can be knowledge-based networks, such as STRING [30], data-driven, such as correlation networks of multi-omics data from experiments, or a combination of both.

After the computational integration step, other algorithms or visualizations can be applied. For example, the whole repertoire of network algorithms can be used for analyzing composite networks. Furthermore, visualization is a indispensable tool for communicating analysis results and forming hypotheses.

2.2 The Fundamentals of Information Visualization

Visualization is the process of bringing information into a visual form, such as an image or a chart to communicate a message. Therefore, the earliest visualizations can be seen as images painted on walls of caves representing objects or situations. The purpose of these visualizations is unknown, yet one can speculate that they *communicate* a concept, such as religion, or that they are meant to be enjoyed. This means that visualization is thousands of years old. In contrast, the field of *information visualization* is much younger. In the 18th and 19th century, pioneers, such as Florence Nightingale and William Playfair contributed groundbreaking visualization techniques, like bar charts, pie charts and line charts before the advent of computers [85]. Nowadays, interactive and static charts can be created automatically dealing with large amounts of complex, high-dimensional information. Here, the main goal is to *explore* data to gain insight. This section highlights why information visualization is necessary, why it is effective, and how visualizations are created.

2.2.1 Perceptual Basics

More research is done in the field of vision than for all other senses combined [86]. A reason for this is that seeing is regarded as the most important since it is seen as the most essential sense and has the biggest proportion of the brain dedicated to it [87]. As discussed by Hutmacher, in reality, it is impossible to rank senses by their importance [86]. Each sense has a high importance for navigating and perceiving our environment. While seeing helps us to find our way around and recognize our friends, hearing is essential for communication, and smelling, tasting and feeling can warn us of dangers, such as fires, poisons, or sharp objects. Moreover, the brain areas dedicated to processing visual information react to a combination of inputs from multiple senses and are repurposed completely in blind individuals.

The frequent usage of visualization as a tool for communication is rooted in other properties. In contrast to the senses of smell, touch, and taste, which are only perceived consciously when the input changes, vision helps us navigate the world by supplying a lot of information in parallel at every moment [88]. Similarly, auditory information is supplied constantly, but while visual input can be static (when looking at a picture or a graph), sound has an inherent temporal aspect, which makes keeping it in working memory hard. Also, cultural factors have to be considered [86]. For example, the "Gutenberg revolution" has shifted the predominant transmission of information from an oral form to a written, visual form [89].

Not all information is kept in memory, but the brain filters visual input to extract important information for our current task [90]. Based on the extracted information, the brain redirects attention to other areas, and then again does the processing. The memory dedicated to solving the current task is known as the *working memory* and the process to solve the tasks is known as *pattern recognition* For example, when searching for a book on a shelf, we (ideally) do not scan all book titles and authors one by one until we have found the right one, but we often employ other search strategies. We might recognize a pattern in the books' order, such as an alphabetical order or an ordering by color. After finding the pattern, our brain shifts the attention to details, by scanning the titles and

authors and matching them with the search query. Once the book is found, no other book is likely remembered although the title and author both were actively perceived.

The human brain has an immense capacity for pattern recognition [91]. The task of visualization designers is to leverage this capacity for data exploration and hypothesis generation by supplying compatible visualizations accounting for the short working memory. Yet, pattern recognition is not only an ability of humans but algorithms are developed to perform this task [92]. They do not have the short working memories of humans and are also more predictable in their interpretation. One might wonder why visualization is even necessary if these other techniques exist. The central difference is, that while algorithms can help answer questions, they cannot define the questions [93]. Visualizations, on the other hand, help form hypotheses. Algorithms do not replace the human in the loop, they only change when humans are useful and necessary in an analysis process [93]. A simple example would be clustering. Of course, one could visualize the attributes of each item in the data and use a human to group the items by similarity. Yet, algorithms exist that can do this much faster and more accurately. Therefore, the clusters are created algorithmically and humans are used to interpret the clusters based on background knowledge.

2.2.2 Channels and Data Types

For effective visualization, it is crucial to understand which patterns are easy for the human brain to recognize and which are challenging. The concept of visual channels helps develop effective visualizations concerning the data displayed. From a perceptional point of view, "Channels are defined by the different ways the visual image is processed in the primary visual cortex." [86]. Channels are used to make objects distinct. For example, a red dot in a set of black dots stands out. Here, the channel used is *color*. Many channels exist, such as color, position, shape, and size.

In her book "Visualization Analysis and Design" Tamara Munzner relates channels to data types and data set types to further discriminate which channels are effective for which type of data [93]. Among

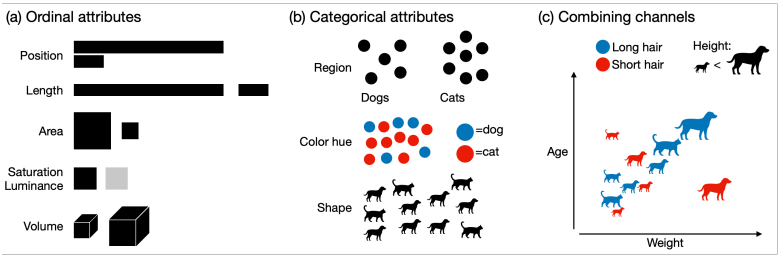


Figure 2.2: Examples of channels illustrating their effectiveness. (a) For the ordinal attributes, the larger attribute value is five times larger than the smaller value. The differences are most effectively encoded by position and length and least effectively by volume. (b) The channels for categorical attributes encode identity in different ways. In this example, the identity of animals is represented. The region channel most effectively encodes the categories, making it easy to, for example, count the number of dogs. Color and shape are less effective. The counting task is hard for shape, as each object must first be identified as either a dog or a cat. (c) Channels can be combined to encode multiple attributes. Here, position, shape, size and color are combined.

the defined data types are *items*, *links*, *positions* and *attributes*. An item is a discrete entity, such as a gene. Links connect items, for example, two genes involved in the same pathway. Positions correspond to spatial 3D or 2D positions, such as the gene's position on the genome. Items, links, positions, and entities of grids can be associated with attributes, which are properties, such as the expression of a gene. Attributes can be categorical or ordered. Categorical attributes have no inherent ordering, such as gene annotations. Ordinal attributes can either be quantitative, meaning that a numerical value can order them, or they can be ordinal, which means that there is inherent ordering but no direct quantification, such as ordering clothes by size labels.

Based on the data types, data set types are defined [93]. They include tables, networks, geometry, and a joint data set type of clusters, sets, and lists. While tables consist of items associated with attributes, networks contain nodes as items, links as edges, and attributes as edge or node values. Geometry relates items to

positions. Finally, sets, lists, and clusters contain items. While sets are unordered, lists are ordered and clusters group items based on specific attributes.

Each item is encoded by a mark, the basic visual element in 2D (or 3D) space. For each mark one or multiple channels encode attributes, thus making the marks distinct. Munzner groups channels by their expressiveness and ranks them by their effectiveness [93] (Figure 2.2a,b). Expressiveness means that the properties of the data should be reflected in the data, for example, categories consist of distinct items without an ordering, while numbers have an inherent ordering that should be expressed. The effectiveness of a channel depends on multiple factors. A channel should be accurate, which means that the perceived value corresponds closely to the encoded value. Moreover, it also should lead to discriminable objects, and be combinable with other channels.

The effectiveness of channels is extensively researched as discussed by Munzner [93]. Experiments conducted by Cleveland and McGill in 1984 [94] support the ordering by effectiveness. Hutmacher highlights the neurological and psychological basics of the perception of channels [86]. Categorical and ordinal attribute types are encoded using different channels. Munzner rates position as the most effective channel for ordinal attributes, followed by length, area, angle and tilt (as used in pie charts), area, color saturation and luminance, curvature, and 3D volumes. Figure 2.2a provides an intuition for some of the channels' effectiveness. The example shows two attribute values, of which the larger value is five times the smaller value. With the position channel this difference is quite easy to estimate, but it becomes increasingly hard with the other channels. For categorical attributes, the spatial region is ranked best, followed by color hue, motion, and shape. Figure 2.2b illustrates this for region, hue, and shape. While counting the numbers of dogs and cats is easy using the region channel, it is harder when they are discriminated by color, and it is hardest when shapes have to be compared. However, the region channel is hard to combine with other channels. For example, with the combination of channels displayed in Figure 2.2c, the position of an item in a 2D space also encodes ordinal attributes. Therefore, the region channel is no longer available

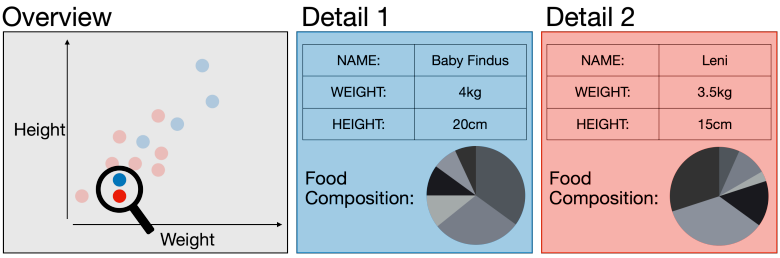


Figure 2.3: An example of a dashboard with an overview-detail setup and small-multiples illustrating "Shneiderman's mantra" [95]. The overview visualization shows the height and weight of dogs and cats in a scatterplot. A set of dots can be selected and further information on the pets' food compositions is displayed in detailed views corresponding to small-multiples.

for encoding categorical attributes, which are encoded using color and shape instead.

2.2.3 Combining Channels

Based on the channels described by Munzner, different plot types can be created, as demonstrated by the scatterplot in Figure 2.2c. When designing a plot, the most effective and expressive channels should be chosen. According to Edward Tufte, it is important not to add any other "non-data-ink" to the visualization, as it might distract the viewer from the actual data [96]. The concepts of a high data-ink ratio and avoiding unnecessary elements called "chartjunk" are useful guidelines when designing a visualization. Yet, the visualization literacy of the viewers must also be taken into account. While minimalist designs have the best data-ink ratio, they can be hard to understand as they are not what the viewers are used to [97]. Furthermore, non-data items in charts can make them memorable [98].

Due to the high number of channels, there is an enormous number of possible plot types. Here, the most common ones for visualizing multi-omics data used in the approaches presented in this dissertation are described. The applicability of a plot depends on the type

Table 2.1: Types of plots used in this dissertation

Data Set Type	Plot	Items	Attributes	Links
Table	Line chart	x	x	
	Bar chart	x	x	
	Heatmap	x	x	
	Radar chart	x	x	
	Data table	x	x	
	Boxplot		x	
	Ribbon chart		x	
	Histograms		x	
Sets	Venn diagram	x		
	UpSet plot	x		
	Parallel Sets / Sankey diagram	x	x	
Networks	Node-link diagram	x		x
Trees	Dendrogram	x		x
	Treemap	x	x	x

of the data set (Table 2.1). Multi-omics data is high-dimensional and encompass many of the proposed data set types. Genes have a geometrical property, which is their position on the genome. Abundance data, such as transcript or protein quantification data is tabular, where each row corresponds to a gene (the item) and each column corresponds to an attribute, which can be the gene description or quantitative abundance measurements at specific conditions. Integration methods are applied that produce other types of data sets. Based on the tables, composite networks can be created, such as gene co-expression networks. Similarly, associations between proteins are studied in Protein-Protein Interaction networks. Finally, oftentimes items, such as genes, are extracted from tabular data based on properties, such as similar abundance values, leading to sets of items, clusters, or lists of items ordered by a quantitative value.

For each of the data set types, specific plot types are commonly used for visualization. Table 2.1 shows an overview of plot types applied in this thesis and relates them to data set types, items, attributes, and links. Tables can be visualized using well-known plots like line charts, bar charts, or heatmaps. Bar charts and line charts encode items and their attributes using position in two dimensions. While bar charts visualize items using separate bars, for example showing the expression value of different genes, in line charts the data points are connected, e.g. to show the development of gene expression over time. This means that line charts are more expressive for ordinal data and bar charts are better suited for categorical data. Heatmaps encode attributes using color luminance or saturation [99]. They can be seen as a table that displays quantitative values, where the value is indicated by the intensity of the color. Heatmaps can show many items at once, but there is a trade-off regarding effectiveness, as color is less efficient for encoding quantitative values than position. A variation of the described charts is to place them on polar coordinates instead of linear coordinates. A radar chart corresponds to a plot that contains a polar axis and a radial axis instead of two orthogonal axes. Often, circular layouts are considered inferior, as interpreting positions is harder on a radial axis [96].

Some plots summarize attributes instead of visualizing individual items. Box plots, histograms, and ribbon charts visualize distributions. While boxplots only show specific calculated values of a distribution (median, quantiles, min, max), a histogram shows the number of items per bin, leading to a more accurate representation when the distribution is nonnormal. Ribbon plots visualize the medians of multiple sets of values using a line with a ribbon encoding the standard deviation.

Set visualizations include Venn diagrams, UpSet plots, and Sankey diagrams. Venn diagrams show the intersections and sizes of sets effectively for up to three sets. For more sets, UpSet plots have been developed, which show sets and their intersections using a combination of bar charts and a matrix [100]. Rows of the matrix correspond to sets, and columns to intersections. For each intersection, the cells in the matrix indicate which sets are considered. The bar charts on

the columns show the size of the intersection, while the bar charts on the rows show the size of the sets. Parallel set diagrams visualize pairwise intersections of sets using parallel axes. Each axis represents an attribute, such as gender and age groups of a population. Bands connect the different attribute values showing the shared proportion, such as the proportion of females in the age group 20-30. Similar visualizations are Alluvial and Sankey diagrams. In contrast to Alluvial charts and parallel sets, Sankey diagrams are usually defined as visualizing directional flows, similar to flowcharts. Yet, the differences between these plots are not well defined, therefore they are used interchangeably in this thesis.

For networks commonly node-link diagrams are used where items are visualized as a shape (usually a circle) and lines connect linked items. Similarly, trees are visualized, but the hierarchical structure allows a more ordered placement of nodes and edges in dendrograms, oftentimes used to show phylogenetic species trees. Attributes can be mapped to the edges and nodes of trees and networks, but are not considered a native property of the plots. Treemaps show hierarchical structures together with a quantitative attribute encoded by the area of rectangles. Usually, they show part-of-a-whole relationships, such as the number of species in different taxonomic families and orders.

Oftentimes, data to be visualized consists of multiple attribute and scales. Yet, the number of channels that can be used in a single plot is limited and some channels cannot be effectively combined [93]. Therefore, often multiple plots are combined into one visualization. In *small multiples* the data is divided into groups and each group is visualized with the same encoding, such as displaying maps with election results of different years, or, as in Figure 2.3, visualizing the food composition for two pets. Moreover, other setups are used where plots differ in channel composition and scales. One of the most important setups is combining overview visualizations with detail visualizations. In the overview, the entire data set is visualized. Commonly there is some form of navigation with which data of interest can be selected to be displayed in detail. For example, in the visualization displayed in Figure 2.3 dots in the scatterplot could

be selected by clicking, and detailed visualizations show further information about the dots' identities. This setup relates strongly to the famous concept called "Shneiderman's mantra" which states "Overview first, zoom and filter, details on demand" [95].

2.2.4 Visualization Tasks

When designing a visualization one always has to consider the purpose and the users of the visualization. While the concept of visualization tasks is discussed in Tamara Munzners's book, it is detailed in her collaboration with Matthew Brehmer. Here, they introduce the why-what-how framework: "why the task is performed, how the task is performed, and what are the task's inputs and outputs" [101]. The inputs for the "What" part can be seen as the data types and data sets described in subsection 2.2.2. The output is new data, if the task was to produce, or abstract output, such as "insight" or "joy". "How" describes the described methods of encoding data, manipulating the visualization, and introducing new elements to the visualization. While encoding is done using the techniques described in subsection 2.2.2, manipulating corresponds to interactive methods, such as navigating the visualization, filtering, and aggregating elements. New elements can be introduced by, for example, manual annotation, importing data, or deriving data such as clusters.

Yet, before decisions on how to perform tasks can be made, the goals of the users have to be identified, corresponding to the "Why" part. The goals can broadly be classified into "consume" and "produce". Consume means that the visualization is used to discover, present, or simply to enjoy information. Produce, on the other hand, has the goal of creating new information by annotating data, recording the exploration process, or deriving data. Oftentimes, these goals go hand-in-hand. Every user prefers an enjoyable experience, though in biomedical visualization this is usually not the main goal. Oftentimes, the results of an analysis are presented, but also offer the option for exploration and deriving data for subsequent analyses.

These high-level objectives can be decomposed into more fine-grained terms. Here, the authors distinguish between search and query. For all objectives, users have to search for elements of

interest, such as the books in a shelf, and query the found values, i.e. retrieving information from the books. Search includes looking up, locating, browsing, and exploring objects. Lookup means that we already know where a object is found and we only need to navigate to it, for example, when looking up information in the only book on our desk. Locating refers to unknown targets, like locating a specific book in a shelf. Browsing means that a user is not searching for a specific object, but wants to find objects with specific attribute values, for example, finding books related to information visualization in a library. Finally, exploring relates to a search process without having a specific object or attribute in mind, e.g. when spending time in a book shop without looking for a specific genre or book. The found objects are queried, which means that the attributes are identified, multiple attributes are compared, or all found attributes are summarized.

Naturally, the concept proposed by Munzner is only one way to abstract and encode data. Ultimately, each proposed concept provides systematic vocabulary and mental models for designing visualizations. In chapter 3, a State of the Art Report is described, which specifically adapts Munzners topology for genomic data [24]. Moreover, the application of the adapted topology for the development of a visualization approach which we called `SeMa-Trap` is detailed [8].

2.2.5 The Visualization Pipeline

The concepts presented help design visualizations if the objectives of the potential users are known. Yet, the goals of the users are not always well-defined. Fisher and Meyer present steps for developing visualizations together with domain experts [102]. First, visualization stakeholders must be defined, which means that the groups of people targeted by the visualization must be identified. This can include, among others, analysts, data producers, and decision-makers. In bio-medical visualization, stakeholders can be, for example, bioinformaticians and domain experts analyzing data.

To find the stakeholders' goals communication is key. Interviews can be used to identify the goals and translate them into concrete tasks.

A common technique is semi-structured interviews where some questions are predefined but essentially the interviewer and interviewee explore the visualization problem together. Broad questions concern project goals, properties of the data, expected results, and possible stakeholders. During the interview process, more and more fine-grained questions can help define the tasks. In addition, contextual interviews can be conducted. The goal of these interviews is to observe the stakeholders exploring the data with existing approaches and asking questions about the current issues.

The other main pillar for designing a visualization is understanding the data. The data should always be explored using, for example, Excel, R or Python Scripts. Based on this exploration designers can decide if visualization has to be created from scratch or if an off-the-shelf tool can be used. When the decision has been made that a novel tool has to be implemented, rapid prototyping can be performed to explore ideas quickly. This ranges from manually drawing visualization using paper to creating mockups using, for example, powerpoint slides to implementing an interactive prototype.

Finally, prototypes have to be presented to stakeholders to gain feedback. Importantly, it is not sufficient to simply ask experts if they like the prototype, as this oftentimes produces positive feedback out of friendliness or thankfulness. It is better to again perform contextual interviews and observe stakeholders while they are using the prototype.

In general, these steps can be repeated multiple times. Prototypes are presented to stakeholders, evaluated, and redesigned. Oftentimes, the process starts with simple prototypes which become more interactive and advanced at every iteration.

2.3 Visualization of Multi-Omics Data

A multitude of approaches have been developed for visualizing integrated multi-omics data using either knowledge-based, data-driven, or composite network integration methods. For knowledge-based integration genome browsers such as `IGV` visualize multi-omics data by mapping it to genes [18]. Similarly, `Vanted` [19] and `Pathview`

Web [20] visualize multi-omics data by mapping it to existing molecular pathways, which can be seen as networks from a visualization perspective. Abundance values are often mapped to color scales. Multiple conditions can be displayed by showing multiple tracks in IGV, or by visualizing them as an array of colored squares in the pathway network visualization. While these approaches provide an intuitive mapping of the data, they can only visualize a small number of genes in a genome or nodes in a pathway. Visualization methods applying more advanced integration methods to process the data before visualization are required for finding interesting features and providing overviews.

Paintomics is an approach applying knowledge-based integration and creates a multi-omics overview using pathway enrichment methods for multi-omics data [23]. For visualization, it creates a pathway network, where significantly enriched pathways represent nodes and edges that connect similar pathways. A pathway diagram can be drawn for each pathway, where the abundance values are visualized using color in an array of squares. Furthermore, the major trends for each omics layer in the pathway are visualized with line charts. Due to its knowledge-based approach, its applicability depends on the data being well-annotated and represented in the databases used. Thus, it is usually not applicable for non-model organisms. Furthermore, the actual abundances of the omics layers across conditions are only visualized in the detailed visualization, thus, their impact on the integration result is hidden.

As mentioned, composite network approaches enable the application of graph-based methods on multi-omics data. STRING is a database for a network of protein-protein associations created using multi-omics data, such as genomic neighborhood, co-expression, and literature co-occurrence [30]. It supplies a node-link diagram for a set of proteins and their associated neighboring proteins and details the type of association using differently colored edges. However, it does not offer an overview visualization for the entire network.

The strengths of the different integration methods can be also be combined. 3Omics combines data-driven integration, composite networks, and knowledge-based integration for transcriptomics, proteomics, and metabolomics data [21]. It requires abundance data

from each omics layer as input, where the conditions must be consistent across layers. For data-driven integration, it visualizes a co-clustering of multi-omics data in a heatmap, where the columns correspond to conditions and the rows correspond to omics entities. Furthermore, it supplies a node-link diagram of a composite network, visualizing the correlation between the omics entities. When an omics layer is not contained in the input data, it can also visualize literature-derived relationships as edges, representing a knowledge-based approach. In addition, it performs enrichment analysis and visualizes the significance using bar charts. The network visualization of `3Omics` is limited to displaying a few hundred nodes effectively without too many overlapping nodes, edges, and labels (known as a hairball).

For visualizing data on a larger scale other approaches have been developed, for example, studying diseases using human cohorts. The Genome Browser `Xena` is focused on visualizing multi-omics data for large cohorts [103]. While it can visualize large cohorts and multiple variables, it does not offer a way to find features of interest. Similarly, the cBio portal, a platform for cancer genomics data, visualizes an overview of metadata but does not automatically highlight interesting multi-omics features [104], [105].

This dissertation presents approaches for the integration and visualization of multi-omics data. The methods aim to fill the gaps in the previously mentioned approaches. `SeMa-Trap` presented in chapter 3 applies prediction methods to detect features of interest in a genomic sequence using a knowledge-based approach [8]. It visualizes gene expression of the features of interest on top of genome coordinates, similar to `IGV`. Chapter 4 represents a knowledge-based approach applying statistical methods for calculating lists of over-represented GO terms and visualizing them using a dendrogram and treemaps [26]. Data-driven integration is exemplified in chapter 5 and chapter 6. `OmicsTIDE` presented in chapter 5 applies clustering to proteomics and transcriptomics data and summarizes the clusters using ribbon charts and their overlaps using a Sankey diagram [27]. `OncoThreads` (chapter 6) visualizes patient cohorts associated

with multi-omics data over time [28]. It allows the interactive creation of patient groups according to multi-omics features and visualizes the development of groups over time using a Sankey diagram. Furthermore, it offers methods to find features of interest. Chapter 7 explores how to visualize large composite networks effectively. Here, network aggregation techniques using ego-graphs are applied to reduce the size of the network [29]. The visualization combines conventional node-link diagrams with set visualization.

Chapter 3

Visualizing Genomes

Many multi-omics features, such as nucleotide mutations, gene expression, and protein abundances are associated with one or multiple locations on a sequence. Therefore, visualizing data mappable to a genome is one of the most straightforward ways for multi-omics integration. Moreover, elements of other omics layers are indirectly associated with genomic locations. For example, bacterial metabolites are associated with metabolic gene clusters in bacteria. Researchers study these gene cluster structures and the interplay of different multi-omics data when trying to find the conditions triggering the expression of specific genes in the genome leading to the production of a compound of interest.

A multitude of tools exist for the visualization of genomic data. Section 3.1 summarizes a *State of the Art Report* (STAR) on genome data visualization I contributed to in 2019 [24]. In the STAR, we show, that the high number of tools is not necessarily reflective of a high number of visualization tasks and techniques, but originates in a multitude of data formats and non-systematic ways of tool development. The systematic approach of categorizing genomic data, tasks, and techniques presented in the report can be used for more strategic development of tools. Section 3.2 highlights how the insights of the report have been leveraged for developing the visualization tool `SeMa-Trap`, a tool for visualizing gene expression of biosynthetic gene clusters.

Furthermore, the STAR is the basis of `Gosling`, a visualization grammar for genomic visualizations developed by L'Yi *et al.*[25]. As a more general solution for tool development consisting of genomic and non-genomic visualizations, I developed an extension called `Gosling-Meta`. Section 3.3 describes the development of this

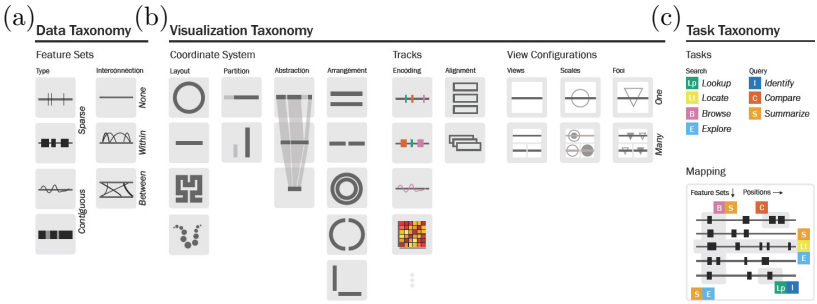


Figure 3.1: Data, visualization, and task taxonomy for genomic visualizations [24].
(a) Genomic data consists of feature sets and is grouped by type (sparse or contiguous) and interconnections (none, within a feature set, between feature sets). (b) Visualizations are categorized by the coordinate system, including the layout of the genomic axis, the partition of the axis, the type of abstraction (hiding/condensing sections), and the arrangement of multiple genomic axes. Tracks on genomic axes display different visual encodings, such as colors and positions. Tracks can either be stacked or overlaid. Furthermore, genomic visualizations have view configurations, determining the scales and different points of focus available for synonymous visual exploration.
Source: Reprinted from the open access article [24] under license CC BY 4.0 `https://creativecommons.org/licenses/by/4.0/`.

extension and shows how this grammar can be used for systematically creating interactive applications by implementing visualizations similar to existing approaches such as `IslandViewer4` [106] and `GeCoViz` [107].

3.1 Tasks, Tools, and Techniques

For the STAR I developed a classification of genomic visualizations together with Sabrina Nusrat and Nils Gehlenborg [24]. In this review, we created taxonomies for the data, tasks, tools, and techniques (Figure 3.1) based on an extensive survey on approaches for genomic visualizations. We restricted the scope to cover only genomic visualizations that visualize data on one or multiple genomic axes. This excludes visualizations like clustered heatmaps (for example, to show gene expression) or node-link diagrams for network visualizations (showing e.g. gene co-expression).

The following sections outline the data, visualization, and task taxonomy. The application of the data and visualization taxonomy to an approach is exemplified for `IslandViewer4` [106] displayed in Figure 3.2. `IslandViewer4` is a tool for comparing the results of prediction methods for genomic islands, which are sequences introduced into bacterial genomes by horizontal gene transfer. In contrast to many other approaches for prokaryotes presented in the STAR, `IslandViewer4` is highly interactive and covers many aspects of the presented taxonomies. A taxonomically annotated cutout can be seen in Figure 3.2a.

3.1.1 Data Taxonomy

The data taxonomy describes the possible setups of genomic data (Figure 3.1a). Elements on a genome are referred to as features, such as genes, mutations, and rearrangements. A feature is associated with a genomic position, a genomic range, and a single or multiple attributes. A set of features of the same kind is referred to as a feature set, for example, all genomic islands. Feature sets are characterized by their type, which is either a point feature or a range feature. Moreover, feature sets cover the entire genome (contiguous

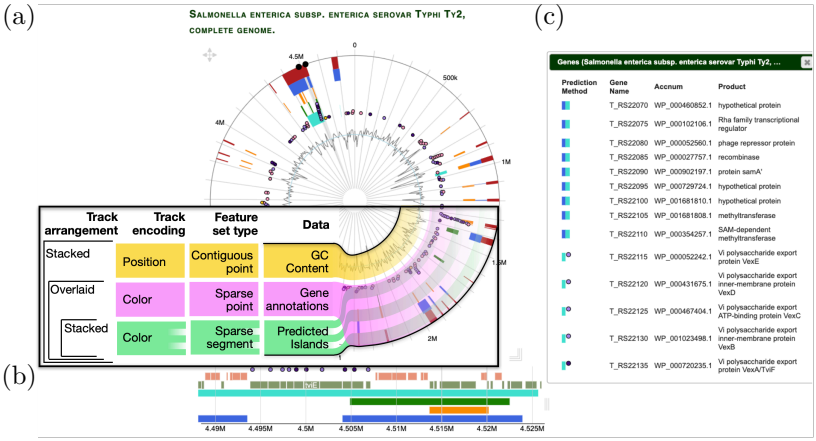


Figure 3.2: Annotated screenshot of `IslandViewer4` [106]. (a) Annotated main view.
Data: The circular overview of `IslandVierer4` shows the predicted genomic islands ●, gene annotations ●, and the GC content ●.
Feature set type and track encoding: The GC content ● is a *contiguous point* feature set. The track encodes the GC percentage as a *function* track. The gene annotations ● are associated with the first position of each gene and are thus *sparse point* features with categorical attributes encoded using *color*. The predicted Islands ● are *sparse segments* with categorical attributes representing the prediction method, encoded using *color*.
Track arrangement: The island tracks ● are *stacked* and the gene annotation track ● is *overlaid*. The GC content track ● is *stacked* in the center. A brush selects a range for the detailed view. (b) The detailed view shows the positions and extent of genes on the plus and minus strands as additional stacked tracks beside the islands and annotations. (c) The table shows detailed information on the genes within the selected range.

features) or only partly (sparse features). Furthermore, feature sets can be interconnected, e.g. when two point features are close in the 3D structure of the genome, or when orthologous genes in multiple genomes are studied. Moreover, feature sets can be associated with metadata. For example, a feature set containing gene expression can be associated with an experimental condition.

The genomic islands in `IslandViewer4` are sparse segment features (Figure 3.2a). The gene annotations are associated with the first position of the genes and are thus sparse point features. For the GC content in the center, each position in the genome is associated with an individual value, thus it is a contiguous point feature set.

3.1.2 Visualization Taxonomy

Data, biological questions, and preferences determine which type of visualization is applied. Visualizations can be categorized by multiple factors as shown in the Visualization Taxonomy in Figure 3.1b.

Visualizations differ in their genomic coordinate systems, which are mostly *linear* or *circular*, but any type of sequential layout can be used. Furthermore, chromosomes (or contigs) can be positioned *segregated* or *contiguous* as if they were a single sequence. The coordinate system can be kept as in the original sequential nature, or *abstracted*. Abstraction means that some parts or all parts of the coordinate system are abstracted to a constant length. For example, *partial abstraction* is often applied when analyzing eukaryotic genes. Introns are sometimes displayed as gaps with a constant size, while exons are displayed with their actual length. In this case, *complete abstraction* would mean that both introns and exons are displayed as blocks of the same size, which would indicate that their length is not relevant to the visualization.

For visualization, data are mapped to the coordinate system. Different feature sets are visualized as different *tracks*. A wide array of visual encodings can be used depending on the data displayed. Color is commonly used to visualize numerical or categorical values. For quantitative attributes, an additional axis can be introduced for visualizing by position, for example when visualizing GC content per position. Tracks can be either stacked, which means that each track

is on a separate lane, or they can be overlaid on top of each other. Stacking tracks increases the necessary space while overlaying can lead to visual occlusion.

A set of tracks aligned to a coordinate system is called a *view*. The setup of views in a visualization is called its configuration. While single-view visualizations only allow visualizing a single genomic window on a single scale at a time, multi-view visualizations can include multiple scales (*multi-scale* configuration) or multiple genomic windows (*multi-focus* configuration).

Furthermore, not only genomic views are of interest. Utility visualizations show metadata on the genomic visualizations. They can either be strongly linked, like metadata aligned directly to tracks with additional information on the feature set, or weakly linked, such as tables showing additional information.

Figure 3.2 illustrates the application of the visualization taxonomy to `IslandViewer4` [106]. A circular layout has been used for an overview visualization (Figure 3.2a) and a linear layout has been used for the detail visualization (Figure 3.2b). The approach encodes the different prediction methods (sparse segments) and gene annotations (sparse points) using color (Figure 3.2a). The GC content track visualizes values on an additional vertical axis (contiguous points).

The approach uses stacked and overlaid tracks (Figure 3.2a). It displays the different prediction methods in five stacked tracks with an overlaid track for the gene annotations. The GC content track is stacked with the other tracks. Since the dots in the overlaid gene annotation track are very sparse features, occlusion is limited and only occurs if many gene annotations can be found in a small window. By default, `IslandViewer4` has two genomic views in a multi-scale arrangement with a view showing the whole genome (Figure 3.2a) and a detailed view (Figure 3.2b). A further genome can be added for visualization. If the same genome is visualized twice, the arrangement is multi-focus, since different parts of the same genome can be explored in parallel. In an additional non-genomic view `IslandViewer4` displays a metadata table that shows further information on the genes visible in the detailed visualization (Figure 3.2c).

3.1.3 Task Taxonomy

In the STAR the task taxonomy developed by Brehmer and Munzner has been adapted for genomic visualizations [101]. Tasks are grouped into high-level and low-level tasks. High-level tasks relate to biological questions, such as "Does bacterium X have many genomic islands?" or even more specifically "Is there a genomic island containing resistance genes that could explain why bacterium X is resistant to antibiotic Y?". Low-level tasks model the tool's interactions and are grouped into "Why?", "How?", and "What?". "Why" refers to the low-level motivation behind the task, e.g. identifying a gene's annotation within a genomic island. "How" refers to the interactions needed to perform the task, for example navigating to the Island of interest. "What" refers to the input and output of the task, such as the gene name as input and its annotation as output.

The why tasks are grouped into search and query tasks (Figure 3.1c). While a search task refers to finding a feature of interest, query tasks refer to interpreting the attribute values of the features. Search tasks are *looking up* features of interest if a single feature set is analyzed or *browsing* different feature sets. If more than one position is analyzed features of interest can be *located* or the entirety of features can be *explored*. Query tasks refer to the actual *identification* of the attribute value of a feature, *comparing* attribute values across feature sets or positions, and *summarizing* the insights gained.

3.1.4 Tools for Prokaryotic Genomes

The tools surveyed in the STAR are characterized using the presented taxonomies and grouped by the number of axes, the density of the features, and the interconnectivity of the features. Of all surveyed tools, prokaryotic visualization has been the minority.

Strikingly, prokaryotic tools are mostly grouped into tools with a single genomic axis and sparse, non-interconnected features. Due to the size of the prokaryotic genomes, it is possible to visualize them as a whole on a linear or circular axis, for example using genome maps [108], [109]. At the same time, eukaryotic genomes are too large and have too many features for visualizing them as a whole. Therefore, more tools exist for eukaryotic genomes with multiple

scales and focus points [110], [111]. In the survey `IslandViewer4` [106] is the only interactive multi-scale and multi-focus example specifically designed for prokaryotic genomes [106].

3.1.5 Conclusion

The low presence of prokaryotic tools in the STAR report is salient. Research in eukaryotes, such as cancer research receives massive amounts of funding. Therefore, more tools focused on cancer research might be developed and approaches on prokaryotes are less likely to be cited and thus found in literature research.

The visualization approaches for prokaryotes have less advanced view configurations (*multi-scale* or *multi-view*) compared to approaches for eukaryotes. This might be rooted in the earlier sequencing dates of prokaryotic genomes. Prokaryotic genomes are small and thus were the first that were assembled completely. Therefore, many visualization tools were published in the early 2000s and thus, lack interactivity due to technical limitations.

Moreover, less complex genomes, such as prokaryotic genomes, require simpler visualization. Prokaryotic genomes are structured differently from eukaryotic genomes. While prokaryotes have a high density of genes, eukaryotes have exons, introns, and sequences of unknown function, often not considered in analyses. Thus, visualizations for eukaryotes sometimes hide parts of the genome (abstraction) or consider alternative splicing, while visualizations for prokaryotes usually do neither. Furthermore, eukaryotic genomes often consist of multiple chromosomes. This complicates analysis, for example, when comparing eukaryotic genomes of different species, since orthologous sequence regions can be found on different chromosomes. Moreover, prokaryotes have more operon-like structures, which means that a set of genes of interest is often in spatial proximity, requiring visualizations focused on a small genomic window instead of providing means of navigating the whole genome.

Nevertheless, `IslandViewer4` [106] shows that visualizations with advanced view configurations can help enhance prokaryotic analysis. The tool and its previous versions, of which the first one was published in 2009, are highly cited and widely applied. In principle,

this type of analysis can be transferred to other groups of contiguous genes, such as biosynthetic gene clusters.

3.2 SeMa-Trap

The insights gained in the STAR have been used for developing `SeMa-Trap`, a tool for visualizing *biosynthetic gene clusters* (BGCs). It consists of a computational pipeline and interactive visualizations. The computational pipeline was developed in the Ziemert group by Mehmet Direnç Mungan. I conceptualized and implemented the visualizations and discussed them in group meetings with our collaborators from the Ziemert group and other domain experts. Both the pipeline and the visualization were published in an article in Nucleic Acids Research in 2021 [8]. While the publication contains sections on the visualization, this chapter describes the visualization and design process in detail. It contextualizes it with my previous work on classifying genomic visualizations [24].

3.2.1 Introduction

Natural products are compounds produced by living organisms. In a more narrow definition in organic chemistry, natural products are *secondary metabolites*. This means that the compound is not essential for the survival of the organism producing it, but might be beneficial. The compounds produced are of interest in many areas, ranging from compounds used in industry and cosmetics to the discovery and production of medical compounds, such as antibiotics [112], [113].

In bacteria secondary metabolites are usually produced in *biosynthetic gene clusters* (BGCs), operonic structures including most of the genes involved in the production of the compound. BGCs are only expressed under specific conditions since secondary metabolites are not essential to the bacterium's survival and are only produced if needed. Antibiotics are often produced as secondary metabolites, for example, in response to competition with another bacterium [82]. Therefore, finding BGCs producing compounds of interest and

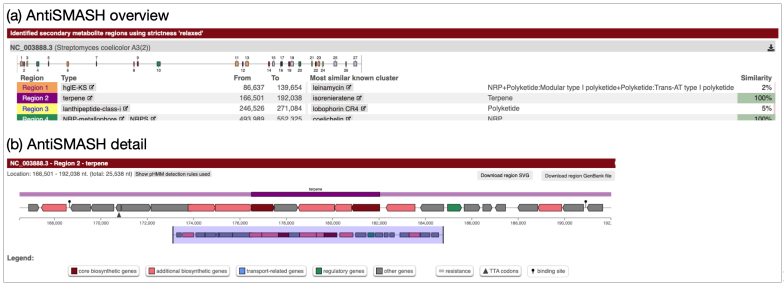


Figure 3.3: Screenshot of `antiSMASH` [63], [115] visualizations (a) Excerpt of the `antiSMASH` overview visualization showing the genomic location and extent of the BGC regions. The BGC regions are listed in a table as rows, columns include the BGC region number, the type, the extent, the most similar known cluster, and the similarity value. (b) Detailed visualization of `antiSMASH`. The cluster structure is visualized on a genomic axis. Triangle shapes indicate gene orientation, color indicates gene type.

triggering the conditions necessary for their production is of interest in discovering novel antibiotics [114].

Whole bacterial genomes can be scanned for BGC regions using bioinformatics software, such as `antiSMASH`, an established pipeline for predicting and visualizing predicted BGC regions of a reference genome [63], [115]. The position and extent of predicted BGC regions are visualized on a genomic axis (Figure 3.3a). Below the visualization, each region is listed with its extent and most similar known cluster with a similarity value. Furthermore, each region can be inspected in detail, showing the genetic structure including the genes' position, extent, and annotations (Figure 3.3b). Further visualizations and tables show information on specific domains, detailed information on genes, and similar cluster structures on other organisms. A set of pipelines and tools have been developed based on `antiSMASH`. `IMG-ABC` is a BGC database that uses `antiSMASH` for BGC prediction and visualization of clusters [116]. Furthermore, it includes a separate BGC neighborhood viewer that visualizes the genes beyond the borders of the BGC. `antiSMASH` is a purely genomic tool focused on predicting the structure of a BGC. However, it does not include other experimental data such as expression data.

Heatmaps are one of the most common visualizations for expression data. Usually, gene expression heatmaps are clustered hierarchically facilitating the identification of co-regulated groups of genes. However, when visualizing the expression of operonic structures in bacteria the inherent structure of the operon genes is of interest. Therefore, tools like `Genome2D` [117] and `MINOMICS` [118] visualize genes on the genomic coordinate system encoded as arrows colored by gene expression using a color scale indicating up-regulation (green), no change (gray), and down-regulation (red). `MINOMICS` visualizes multiple conditions using one line per experimental condition. In addition, it shows additional annotations, such as motifs or translation stop sites.

While these approaches highlight common routes for visualizing BGC expression, no native approach exists combining the predictions made using `antiSMASH` with analysis and visualization of differential expression and co-expression. This chapter presents the `SeMa-TRAP` visualization approach for exploring BGC expression and co-expression and showcases how the tasks, data, and techniques can be abstracted for visualization design using the STAR [24].

3.2.2 Pipeline

The `SeMa-Trap` pipeline computationally predicts BGCs, quantifies gene expression, and processes the data for visualization. This section provides an overview of the pipeline with the information relevant to the design of the visualizations. I refer readers to the original publication for details on the data formats and the bioinformatics tools.

The pipeline input consists of RNA-Seq reads, a reference genome, and the name of the taxonomic clade of the bacterium. The pipeline predicts BGCs using `antiSMASH` [115], annotates genes involved in BGC expression, and identifies housekeeping genes. Moreover, the potential metabolite produced by each BGC is predicted. For RNA-Seq data analysis the pipeline maps the RNA-Seq reads to the reference, counts the reads mapping to each gene, and calculates differential expression.

To find potentially concordantly or discordantly regulated genes and BGCs a *regulation association score* dependent on gene expression is calculated between each BGC and each gene. The fold changes of the gene and the BGC are multiplied and added up for each condition. High scores indicate that a gene's expression could be (concordantly or discordantly) associated with the expression of the genes on the BGC. However, it is only reliable when the expression is studied across many conditions.

In addition, the pipeline calculates the expression of BGCs relative to a reference set of genes. Thereby, users can deduce if the BGC is sufficiently expressed for their purposes. As there is no defined threshold for when a BGC can be considered expressed, the pipeline calculates the average expression of all genes, housekeeping genes, and non-housekeeping genes. The expression of housekeeping genes represents a conservative threshold for BGC expression, as they are usually expected to be highly expressed. Yet, even if a BGC's expression is below the expression of housekeeping genes, the expression can be sufficiently high.

3.2.3 Visualization Design

For the `SeMa-Trap` visualization, we leveraged the insights gained in the STAR [24] by systematically designing a visualization. The process starts with a requirement analysis based on the data and tasks. The requirements determine the selection of appropriate and effective visualizations, which take into account domain expert knowledge and expectations.

Task Categorization And Requirement Analysis

During the discussions with the collaborators, we identified high-level tasks for the visualization, as proposed in the STAR [24] and in subsection 3.1.3. The tasks are formulated as biological questions:

Q1 Can I use one of the conditions I applied in my experiment to trigger a relevant expression of my cluster(s) of interest?

Q2 Are there any other clusters than my cluster(s) of interest expressed that I might not have expected? If yes, which ones?

Q3 How are the genes of the cluster expressed?

Q4 Are there any other genes in the genome whose expression correlates with the expression of my cluster?

Q5 What is the annotation of my genes of interest? Which genes have a similar annotation?

First, we grouped the high-level biological questions into detail and overview tasks. Overview tasks involve multiple BGCs, while detail tasks only involve one BGC and its context. Therefore **Q1** and **Q2** are considered overview tasks, while the other three tasks (**Q3**,**Q4**,**Q5**) are considered detail tasks.

Both, the high-level overview and detail tasks can be translated into the low-level tasks suggested in the STAR [24]. Often, domain experts study a specific cluster. Therefore, the visualization should enable them to *look up* their desired cluster and identify its overall expression level (**Q1**). However, if they want to *explore* the unexpected effects of their applied condition, they need to be able to *locate* expressed clusters by *comparing* expression levels (**Q2**). For multiple conditions, researchers should be able to *explore* the different conditions across the clusters. Finally, researchers want to be able *summarize* the effects of the different conditions on the different clusters.

For the detailed tasks, users should be able to *look up* genes of interest, *locate* highly expressed genes (**Q3**, **Q4**) or genes with a specific annotation (**Q5**), *browse* the expression of a gene under different conditions, or *explore* the expression of all genes across conditions. They should be able to *identify* expression values as well as *compare* them across genes or conditions and *summarize* their results, for example when analyzing if the expression across a cluster is constant or not.

In addition to the systematically identified tasks, there are additional requirements for the visualization that involve the tool's relatedness to `antiSMASH` [115] as well as to the shared potential user base:

R1 The visual encodings of `antiSMASH` [115] should be reused wherever possible.

R2 The tool should include visualization and vocabulary for domain experts with little affiliation to computer science and bioinformatics.

Data Categorization

Following the STAR [24], BGC data can be referred to as a sparse feature set, where each gene represents a segment. The BGC can also be seen as a segment feature ranging from the first position of the most 5' gene to the last position of the most 3' gene. The gene features are associated with multiple attributes: The gene's expression per condition, the gene's annotation in `antiSMASH`, and the gene's annotation in other databases. The BGCs are associated with a metabolic product, the genes contained, and their expression. Co-expression can be seen as an interconnection within a feature set.

3.2.4 Design in Taxonomic Context

The visualization consists of two views in separate tabs, one focused on the overview tasks and one focused on the detail tasks. The overview contains summary visualizations for the predicted BGC regions, listing the regions and visualizing their overall expression. Thereby, a region of interest can be *looked up* and the overall expression can be *identified* (**Q1**), or highly expressed regions can be *located* (**Q2**). The detail visualizations are focused on single BGCs and visualize the expression (**Q3**) and annotation (**Q5**) of genes within the region and outside of the region to *identify* co-regulated or inversely regulated genes (**Q4**).

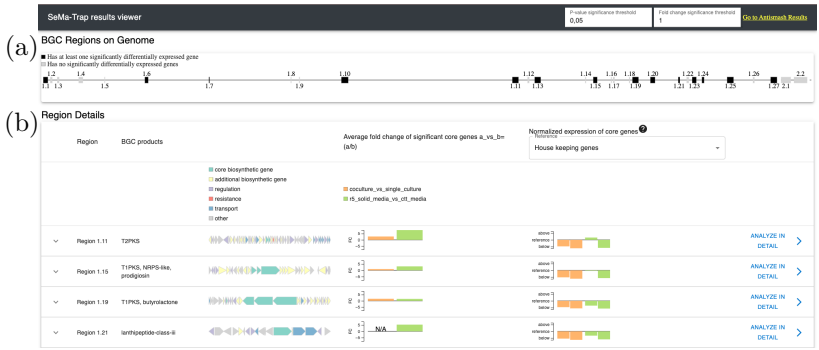


Figure 3.4: Screenshot of the Overview visualization of SeMa-Trap. (a) Visualization showing the genomic location and extent of the BGC regions. (b) The BGC regions are listed in a table as rows, columns include the BGC region number, the name, a visualization of the BGC structure, average expression of core genes, and relative expression compared to other genes.

Overview

In concordance with antiSMASH (**R1**) the overview consists of a visualization showing the positions and size of the BGCs along the genome (Figure 3.4a). The STAR shows that circular layouts are commonly used for visualizing entire circular bacterial genomes or interconnections. Yet, a linear layout similar to the overview visualization of antiSMASH allows space-efficient placement of the visualization at the top of the web page and facilitates label orientation.

The objective of the overview tasks (**Q1**, **Q2**) is to summarize the clusters' expression. Therefore, the main visualization is a table containing information on the BGCs present (Figure 3.4b). The table contains columns showing the BGC region number, its name, and visualizations for its structure, its average expression of the core genes in the conditions studied, and the core genes' expression relative to the mean expression of all or a subset of genes (**Q1**, **Q2**). Each row can be expanded to show a detailed expression of the BGC as a heatmap (Figure 3.5, **Q3**, **Q5**).

The structure of the BGC region is visualized on a linear axis where an arrow-like pentagon or triangle represents each gene if the gene

is below a certain length (Figure 3.4b, 3rd column). The genes are colored according to their classification in `antiSMASH` [115] (see Figure 3.3b for a comparison). Showing the cluster structure can help users identify unexpected patterns (such as missing genes) compared to their knowledge of the cluster studied. Moreover, it serves as a point of recognition for the cluster in addition to the name displayed (**R1**).

The average fold changes of the core genes and the expression relative to other genes are visualized as simple bar charts (**Q1**,**Q2**). The fold change bar charts aim to investigate if a condition applied had a noticeable effect on increasing or decreasing a BGC region's expression. Therefore, a bar in the average fold changes bar chart represents the comparison of two conditions, such as comparing mutants to wild type.

In addition to the average fold change bar plot, the plot visualizing the relative fold change aims to investigate if a BGC is expressed at a relevant level. Users can choose to compare the core genes of the BGC to an expression threshold they consider relevant, including the mean expression and the mean expression of groups of genes like housekeeping genes or non-housekeeping genes. For the visualization of relative expression, every condition is visualized separately. The 0 point of the y-axis is centered on the reference level and the bars indicate if the expression is above or below the reference level. The wording "above" and "below" instead of precise relative expression levels was decided together with our collaborators who are domain experts (**R2**). Precise expression levels require understanding the range of values and the normalization performed, which would call for additional visualizations or explanations that ultimately complicate the visualization. Furthermore, the negative relative expression could lead to confusion, as expression is by definition non-negative.

With an expand icon at the beginning of each row, the row can be expanded (Figure 3.5). For a compact representation of the expression of all genes in a cluster, genes are abstracted to blocks of the same length and the strand information is omitted. Each track shows the expression at a different condition forming a heatmap.

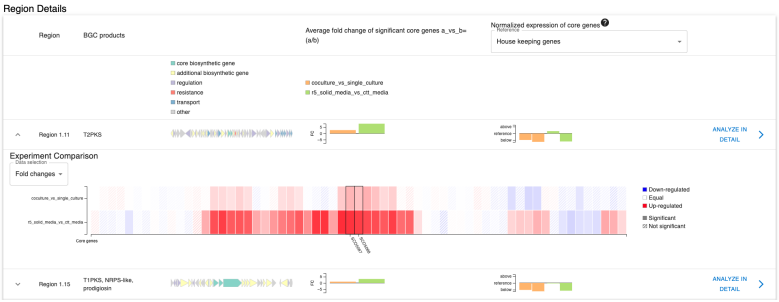


Figure 3.5: Screenshot of expanded row in SeMa-Trap. Each row can be expanded for details on the expression of a BGC region as a heatmap. Rows visualize the different conditions, while columns visualize genes. A color scale from blue to white to red indicates expression. Significantly differentially expressed genes receive a solid fill, while non-significantly expressed genes are striped. Core biosynthetic genes are highlighted with a black frame.

Users can select to show fold changes or normalized expression values. If fold changes are chosen the color scale ranges from blue to white to red. Significantly differentially expressed genes receive a solid color fill while non-significant genes receive a striped fill. If the normalized expression is chosen, the color ranges from white to red (low to high). A black frame around the corresponding column highlights core genes. With the 'analyze in detail' button at the end of each row, users can analyze a BGC in more detail as explained in the next subsection.

Detailed View

In the detailed view, the expression of a BGC is visualized in detail and context with the entire genome (**Q3**, **Q4**). The visualization consists of controls (Figure 3.6a) and three views: A visualization of the BGC (Figure 3.6b), a genome-browser for the entire genome (Figure 3.6c), and tables (Figure 3.6d) showing genes discordantly or concordantly regulated with the BGC core genes.

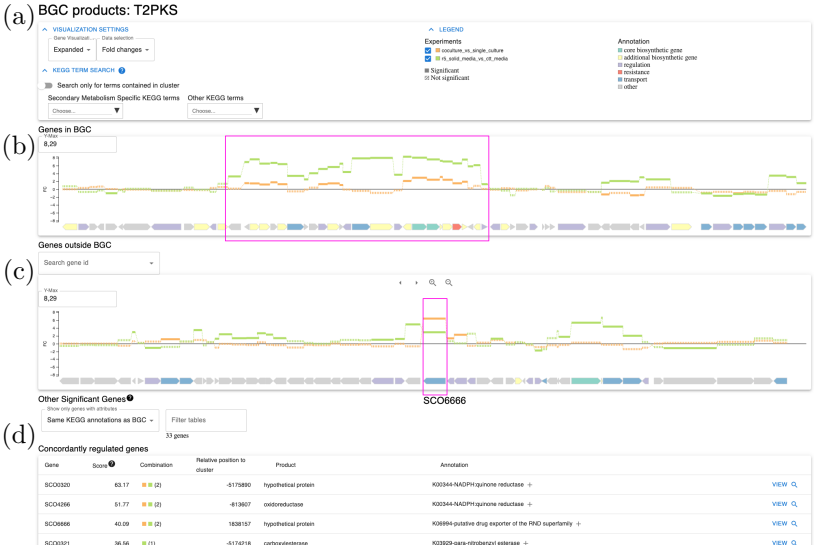


Figure 3.6: Screenshot of the detailed visualizations of SeMa-Trap. (a) Controls provide ways to filter or highlight genes and conditions in the BGC visualizations. (b) The expression of the BGC at different conditions is visualized as a line chart. Differentially expressed genes are represented by solid lines, others by dashed lines. Regions with similar expressions can be identified (pink rectangle). Below, the structure of the BGC region is visualized. (c) A genome browser can be used to view the expression and structure of any region of the genome. (d) A table shows potentially concordantly regulated genes, their *regulation association scores*, the conditions at which they are differentially expressed, and their annotations. Analogously, potentially discordantly regulated genes are displayed in an additional table below (not shown). The concordantly regulated gene SCO6666 is highlighted (pink rectangle).

Visualization controls and search functionalities enable users to change aspects of the visualization and search for functional categories (Figure 3.6a). The controls also serve as a legend for the different conditions and the `antiSMASH` [115] categories of the genes. Conditions can be removed and added to the visualization and `antiSMASH` categories can be highlighted.

For the visualization of individual BGCs, the axis layout should (i) enable stacking and overlaying of multiple tracks (ii) be space-efficient and (iii) support the existing mental model of the genomic sequence of the viewer. We chose a linear layout since it fulfills all of these requirements. While linear and circular layouts allow stacking and overlaying tracks, a linear layout better reflects the linear structure of the BGC.

The expression of the genes is visualized as a line chart along the genomic axis (Figure 3.6b). The conditions are visualized in overlaid tracks with different categorical colors. A solid line indicates significantly differentially expressed genes, while non-significant genes are striped. Stacked below, the visualization of the cluster structure shown in the overview is repeated.

With the controls, users can show genes in their original length (no abstraction) or as blocks of the same length (complete abstraction). By default, genes are visualized with their original length, to increase the recognition of the cluster and to prevent misinterpretation about the length of the gene. However, line charts for genes of different lengths can lead to misjudgment of the fold changes, as the area below the line might be perceived as relevant, while only the position on the y-axis is of interest. Therefore, the fold changes for long genes might be overestimated in comparison to those of short genes. Often coverage, i.e. the number of reads mapping to each position along a genomic sequence is visualized similarly, leading to an additional risk of misinterpretation. Therefore, genes can be abstracted to blocks of the same length, leading to an equal relative area under the line for each gene.

Since task **Q4** requires visualizing genes outside the BGC region, a multi-focus view arrangement is needed. Therefore, a genome-browser-like visualization for visualizing any genome region is displayed below the cluster (Figure 3.6b). The visualization is structured identically to the BGC region view. This visualization shows the genomic region surrounding a gene entered in the search box or selected in the table showing concurrently/discordantly regulated genes (described below). Moreover, it can be navigated to the left and right and zoomed independently, which renders the view configuration multi-scale.

Two identically structured tables show potentially concordantly or discordantly regulated genes (**Q4**, Figure 3.6d). The table shows the gene IDs, the *regulation association score* calculated as described in subsection 3.2.2, the combination of conditions with associated regulation, the position relative to the cluster, the resulting protein product, and the KEGG annotations of the gene. Using the "view" button at the end of each row each gene can be selected for display in the genome browser (Figure 3.6d). By default, the table is sorted by the *regulation association score*, but can be sorted by any column. Furthermore, it can be filtered to only show genes with the same KEGG annotations as the current BGC region or only show a specific `antiSMASH` category of genes (resistance, regulation, transport). A search field can be used using any keyword appearing in any column of the tables.

3.2.5 Implemetation

The data are processed using bioinformatics tools and Python scripts as described in the original publication [8] and subsection 3.2.2. The application consists of a flask server and HTML and Javascript code for the front end. Users can upload their data in a web form and run the pipeline on the server. Each pipeline run is associated with a unique ID supplied to the user which is used for retrieving the processed data after the run is completed. The processed data are injected into an HTML template containing the Javascript code for the visualization. The template is served to the client, enabling users to view the visualization in the browser and to download it as an interactive HTML file. The Javascript

code is implemented using React.js for the application structure, material UI for the application style, and d3.js for creating the visualizations.

3.2.6 Use Case

The following use case showcases how the defined tasks can be addressed with `SeMa-Trap`. For this, we reanalyze a data set by Lee, Kim, Chung, *et al.* [119], which includes a set of RNA-Seq experiments with *Streptomyces coelicolor* to activate the expression of the BGC "T2PKS", producing the antibiotic *actinorhodin*. For this, the authors applied several conditions including an isolated culture of *S. coelicolor* or in co-culture with *Myxococcus xanthus* to mimic the ecological habitat of *S. coelicolor* where actinorhodin is produced due to iron competition. Moreover, different media have been used to study the effect of iron restriction.

Using the `SeMa-Trap` overview (Figure 3.4), we can identify that the core cluster genes of "TP2KS" are differentially expressed and that genes are above the expression of housekeeping genes in the culture on the iron-restricted medium (**Q1**). Using the table, other regions of interest can be located (**Q2**). For example, the regions "T1PKS, NRPS-like, prodigiosin" and "T1PKS, butyrolactone" have differentially expressed core genes. Like "TP2KS" they produce polyketide synthases. Similar products might be produced using genetically similar BGCs. Therefore, the differential expression could arise from sequence similarity instead of a true signal.

As a next step, the gene cluster of interest is explored in more detail (**Q3**) and the detailed view of "TP2KS" is used (Figure 3.6). Using the expression view of the BGC region, we can see that not all genes are differentially expressed but only within a window in the center of the displayed region. Furthermore, the expression is similar for the two comparisons (coculture vs single culture and r5 solid media vs ctt media), apart from a region in the center where the direction of the fold change differs and the differences are only significant for the media comparison. By hovering over the annotation we see that one of the genes within this central region is an activator for the operon (**Q5**). One can speculate that this activator

contributes to the increased expression of the overall cluster in the media comparison.

Genes outside of the BGC region might influence its expression as well. Therefore, as a next step, we want to find genes with similar functions with a similar expression outside the BGC region (**Q4**). For this, we use the table of concordantly regulated genes and restrict the table to genes with the same annotations as in the BGC region. Only three genes are concordantly regulated with both comparisons. Of those, gene *SCO6666* has been shown to affect actinorhodin production in low-iron conditions [119].

3.2.7 Discussion and Future Work

`SeMa-Trap` represents an extension to the popular tool `antiSMASH`. It combines the prediction of BGC regions with expression analysis and visualizes the data on an overview and detailed level. Furthermore, it enables researchers to identify genes potentially co-regulated or inversely regulated with BGCs. The design process of `SeMa-Trap` demonstrates the systematic development of an effective visualization by combining abstraction of the visualization problem with domain expert knowledge.

`SeMa-Trap` is made available to a large pool of researchers due to the coupling of the visualization to an established tool, `antiSMASH`, for BGC prediction and a computational pipeline. Yet, this coupling also puts constraints on the design process of the visualization. Lifting these constraints can make `SeMa-Trap` a universal approach for visualizing BGCs, or other regions combined with multi-omics data.

The design decisions for the overview on the BGC regions (Figure 3.4a) were heavily influenced by the requirement of reusing the visual encodings of `antiSMASH` (**R1**). Like `antiSMASH` `SeMa-Trap` uses a linear axis where the location and extent of the BGC regions are depicted. A more advanced overview could be achieved by placing the BGCs on a circular axis and using the circle space to link BGCs with bands, which could, for example, indicate the calculated score between two BGCs. This circos-like [120] visualization would only be effective when a minimum number of conditions for score calculation is guaranteed.

However, currently, SeMa-Trap is limited to transcriptomic data integration. However, it could be extended to include other omics levels. Mutation data could be included to show the effect of changes on the genome level on gene expression. Furthermore, proteomics data could be included to show how changes in transcript expression are reflected in the proteome. Finally, as BGCs produce metabolites, integrating metabolomic data can help assess the activity of a cluster. In the SeMa-Trap pipeline scores are calculated for identifying potentially co-regulated or inversely regulated genes. With a multi-omics extension, these scores must also be extended.

Currently, the SeMa-Trap visualization is coupled to the computational pipeline for the BGC prediction and expression data analysis. However, SeMa-Trap could be transformed into a stand-alone multi-omics visualization tool for regions of interest. Besides BGCs, other genomic features, such as genomic islands, operons, or syntenic regions are often studied. A generalized version of SeMa-Trap could take a list of regions of interest as input and provide summary and detailed visualizations. The visualization grammar for genomic visualizations Gosling can help create a generalized standalone version of SeMa-Trap. Since the Gosling grammar was still under development while SeMa-Trap was developed, the visualizations were created using d3.js. The alteration steps and extensions necessary to Gosling are described in the following section 3.3.

3.3 Gosling-Meta

3.3.1 Introduction

With the advent of sequencing techniques, a multitude of tools for genomic visualization have been developed. As shown in the STAR by Nusrat, Harbig, and Gehlenborg [24], the visualization concepts of the tools are highly repetitive. They can be decomposed into distinct categories using a taxonomic approach, as described in section 3.1.

The taxonomic approach can be used to summarize the research space of genomic visualization as done in the STAR and systematically design tools using the insights obtained as shown in section 3.2.

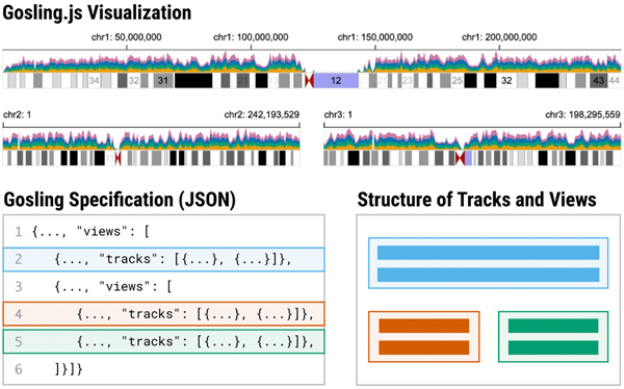


Figure 3.7: Concept of gosling.js [25] ©2021 IEEE. Gosling creates genomic visualization based on a JSON specification. The specification contains the structure of the visualization regarding tracks and views, as well as the data specification and the visual encoding within the tracks.

Furthermore, in the follow-up work of the STAR report, the taxonomy has been used for developing `Gosling`, a grammar for creating interactive, scalable genomic data [25].

With `Gosling`, many different genomic visualizations can be created. The approach is a translation visualization taxonomy presented in subsection 3.1.2 into a visualization grammar that can be used as a Javascript [25] or Python library [121]. Using a JSON format a wide range of genomic visualizations can be created, similar to visualizations created using `Vega` or `Vega-Lite` [122], [123]. In the JSON format genomic views and tracks are defined (Figure 3.7). The data source has to be defined per view or track. `Gosling` can obtain data from simple table-based files (*.csv*, *.tsv*), JSON files, or specific bioinformatics file formats such as *.bed*, *.gff* and *.bam*. The data can be transformed using the grammar, for example, it can be filtered, values can be replaced, or a log operation can be applied to numerical values. For each view, a circular or linear coordinate system can be chosen. Tracks can be overlaid or stacked and different visual encodings can be applied for genomic features. `Gosling` provides options for coordinated views by linking a detailed visualization with an overview.

Non-genomic views touched on in the STAR, are not reflected in the `Gosling` grammar. In the STAR, they are called "Utility Visualizations" which can contain metadata on the genomic views or summarize the content of the visible genomic window. Non-genomic views are especially important for providing context on genomic visualizations, which is necessary for developing fully functional visualization tools. Integrating metadata views into `Gosling` is the main goal of `Gosling-Meta`.

The contributions of this project can be summarized as follows: Literature research was conducted on summary visualizations for genomeic visualizations in prokaryotes to limit the research scope and to fit the prokaryotic focus of the TRR and the previous work on `SeMa-Trap`. Based on this I chose several main visualization types for implementation. A concept for injecting meta-visualizations into `Gosling` was developed jointly with the authors of `Gosling`. Since the published version of `Gosling` was not focused on creating an extensive programmatic API, we extended the `Gosling` API. For `Gosling-Meta` I implemented six types of metadata visualizations. The usability of the metadata visualizations is demonstrated by implementing two examples based on approaches identified in the literature research.

3.3.2 Related Work

Visualization grammars like `Vega` and `Vega-Lite` offer ways to create a large range of interactive visualizations using a declarative JSON syntax [122], [123]. In the JSON, users specify the data, e.g. tabular data, the *marks*, such as bars, and define how the data are translated into *channels*, e.g. the length of the bar. The visualizations are highly customizable and can be combined in multiple linked views. Furthermore, data can be transformed, for example, quantitative data can be binned or filtered based on specified criteria. While `Vega` is more low-level and customizable than Vega-lite, the latter is more concise. Both do not offer genomic views by default.

Yet, the visualization grammars and the STAR have inspired the development of grammars targeted on genomic data, like `Gosling`

and `GenomeSpy`. Similar to `Gosling` `GenomeSpy` visualizes genomic data using JSON specifications [124], [125]. In contrast to `Gosling`, `GenomeSpy` includes a way to visualize meta-data heatmaps in multi-sample visualizations. In the grammar, the metadata and genomic data are stored in two separate ".tsv" files and linked via a sample ID, which is contained in both the genomic data file and the metadata file.

3.3.3 Literature Research

To get an overview of the scope for summary visualizations or metadata visualizations in prokaryotes literature research was conducted using Google Scholar. Tools fulfilling the following criteria were included:

- The publication describes a visualization approach.
- The approach has a prokaryotic focus.
- The approach includes visualization with genome-mapped data.
- The approach contains additional visualizations.
- There is either an illustrative figure or a working example of the approach.
- The publication is written in English and is peer-reviewed.

The first 10 pages of the Google Scholar results were considered. First, general search terms were used such as "genomic visualization bacteria". When a specific topic was identified, such as "comparative genomics visualization bacteria" it was added to a list of topics. For the top five identified topics, further searches were conducted. Table 3.1 shows the search terms used and how many tools were added to the collection in the respective iteration. Terms 1-3 correspond to the broad initial search terms, while the others correspond to specific topics. In total 53 approaches were identified. Additional exclusion criteria were defined based on a detailed review of the approaches. Tools were excluded if genomic and non-genomic views were not linked or not displayed simultaneously or where the linkage

Table 3.1: `Gosling-Meta` search terms used for literature research.

Search Order	Search Term	# Tools added
1	genomic visualization prokaryotes	21
2	genomic visualization bacteria	15
3	sequence visualization bacteria	2
4	comparative genomics visualization prokaryotes	3
5	biosynthetic gene cluster visualization prokaryotes	7
6	secondary metabolite visualization prokaryotes	2
7	genomic island visualization	0
8	CRISPR visualization prokaryotes	1
9	population genomics visualization bacteria	0
10	pangenome visualization bacteria	2

was unclear. After applying these exclusion criteria, 37 approaches remained.

The visualization types were categorized as illustrated in Figure 3.8. The most common visualization type was a table containing metadata (such as `IslandViewer4` [106], see Figure Figure 3.11 for a reimplementation and Figure 3.2 for the original visualization), followed by a phylogenetic tree and a heatmap aligned directly to genomic tracks (such as `GeCoViz` [107], and `GeneSpy` [126], see Figure 3.10 for a similar visualization using `Gosling-Meta`). Different kinds of standard visualizations such as bar charts, pie charts, or histograms were found.

Based on the final collection of visualization approaches, metadata tables, phylogenetic trees, track heatmaps, bar charts, and histograms were selected for implementation. For custom visualizations, we decided to enable the integration of custom `Vega-Lite` specifications.

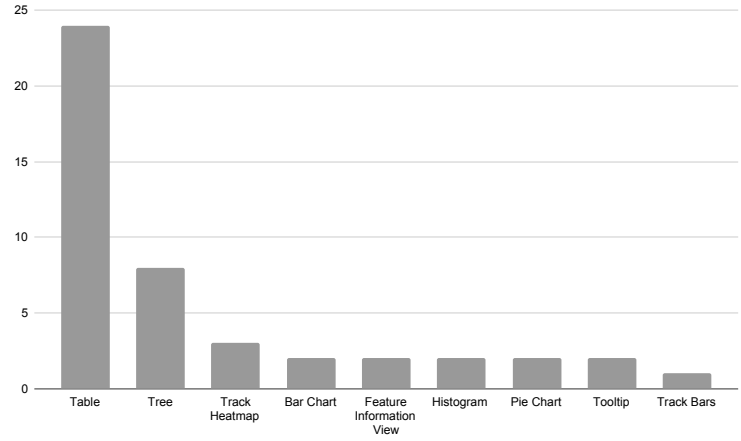


Figure 3.8: Identified Metadata Visualization Types in prokaryotes using Google Scholar. Tables were found to be the most common visualization type, followed by phylogenetic trees and track heatmaps.

3.3.4 Gosling-Meta Grammar

The insights gained in the STAR and the independent literature research for metadata visualizations have been translated into a grammar called `Gosling-Meta`, which extends `Gosling` with metadata visualizations. Like `Gosling`, `Gosling-Meta` is implemented using Typescript and React. For compartmentalization, the code bases and JSON specifications are kept separate. This avoids introducing an overhead of functionality into `Gosling` and preserving its main functionality as a grammar for visualization of genomic data. The `Gosling` specification (in short `Gosling spec`) contains all genomic views and tracks, while the Meta specification (in short `Meta spec`) contains non-genomic views.

Connection Type

In the `Meta spec` the type of the connection has to be defined (`"field": "connectionType"`) specifying how the `Gosling` views and tracks are connected to the `Gosling-Meta` views in terms of

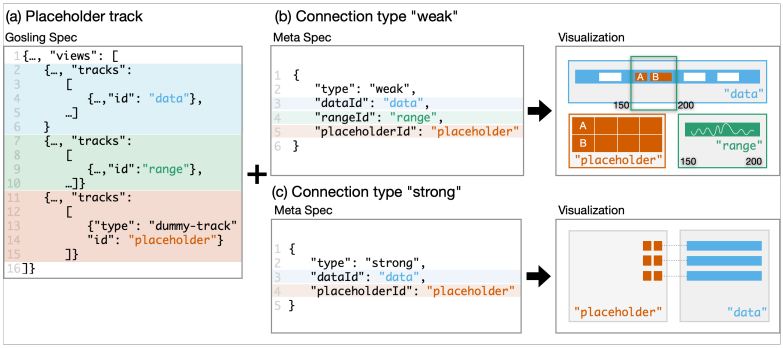


Figure 3.9: Concepts of `Gosling-Meta`. The `Gosling` spec and the `Meta` spec are defined separately. (a) The concept of placeholder tracks is applied in `Gosling` to place `Gosling-Meta` views alongside genomic views. The `placeholderId` is set to obtain the position and extent of the `Gosling-Meta` view. (b) For weakly connected views the `dataId` defines the track displaying the data used for the `Gosling-Meta` view. The `rangeId` defines the genomic range. (c) For strongly connected views the `dataId` defines the multi-row track with which the non-genomic axis is shared.

placement, data, and linkage. For the placement of metadata visualizations within the `Gosling` view structure (depicted in Figure 3.7) the `Gosling` grammar had to be extended with placeholder tracks also called "dummy-tracks" (Figure 3.9a). A placeholder track is an empty track, which can be associated with an ID in the specification. Based on this ID, `Gosling-Meta` can find the position and extent of the track and place the metadata visualization accordingly. In the `connectionType` the `placeholderId` is the ID of the dummy track for the placement of the metadata visualization.

As the STAR suggests, metadata visualizations can be classified as strongly linked or weakly linked [24], which is reflected in the `type` field of the `connectionType` (Figure 3.9b,c). Strongly linked metadata visualizations are directly attached to a `Gosling` track and show metadata on feature sets. This means that metadata visualization and genomic visualization have a shared non-genomic axis. Genomic context visualizations such as `GeCoViz` [107] or `GeneSpy` [126] are strongly linked visualizations where a phylogenetic tree and a metadata heatmap are directly attached to genomic tracks (see subsection 3.3.5 and Figure 3.10 for an example). Weakly linked metadata visualizations show metadata on single features linked via interactions. For example, a metadata table can show details on the data within a genomic window which is updated when the window is moved. `IslandViewer4` [106], presented in section 3.1 and reimplemented in subsection 3.3.5 represents a weakly linked approach.

The data linking between the `Gosling` visualization and the metadata visualization is done using IDs, which are set in both specifications. Often, the same data and genomic range are visualized in the `Gosling-Meta` view and a genomic view (see `IslandViewer4` in subsection 3.3.5). However, the `Gosling-Meta` view can also visualize data from a different source within the linked genomic range, for example, when the `Gosling` view shows genes in a genomic window and the `Gosling-Meta` view shows the mutations within those genes. Therefore two separate IDs specify the track showing the genomic range and the track containing the data. The `rangeId` specifies the `Gosling` track that contains the genomic range, while `dataId` determines the `Gosling` track with the data. For weakly linked views, the `dataId` is used to obtain the data directly from the `Gosling`

track and visualize it in the metadata view (Figure 3.9b). Oftentimes, the `dataId` and the `rangeId` are the same. . For strongly linked views the `dataId` is used for the alignment of the axes of the metadata view and the `Gosling` visualization (Figure 3.9c).

A difference in the `Gosling` grammar compared to the taxonomy in the STAR must be highlighted for strongly linking views. While in the STAR a track is defined as the graphical representation of a single feature set, for example, expression data at a specific condition, `Gosling` introduces the concept of *rows*. This means feature sets of the same type, such as expression data at different conditions, can be visualized as a single track with various rows. This is advantageous for strongly linked metadata visualizations as the order of the non-genomic axis in the `Gosling` visualization can be obtained using the `dataId`.

View Specification

Besides the `connectionType`, each view in the `Meta spec` has a type corresponding to a plot type. Four plot types have been implemented for weakly connected views: `table` for sortable metadata tables, `hist` and `bar` for summarizing quantitative or categorical values within a genomic range in a histogram or bar chart, and `own` for including custom `Vega-Lite` visualizations. Two plot types have been implemented for strongly connected views aligned to multi-row tracks. With `tree` a phylogenetic tree can be aligned to the track and with `track-heatmap` a heatmap-like visualization shows metadata on the different rows.

Furthermore, the data has to be specified, if the views are strongly connected and thus contain metadata about tracks. Similar to `Gosling`, JSON files can be loaded and data transformations can be applied. For example, fields can be merged, or renamed, or numerical fields can be combined using mathematical operations.

3.3.5 Examples

In this section, the usage of `Gosling-Meta` is demonstrated. First, a genomic context visualization is implemented inspired by `GeneSpy` to demonstrate tightly aligned views, phylogenetic trees, and track

metadata [126]. Second, the IslandViewer4 visualization is reimplemented demonstrating weakly linked views and the metadata table [106]. Furthermore, the possibilities of replacing the table with other visualizations are described.

Reimplementation: Genomic Context Visualization

GeCoViz [107] and GeneSpy [126] are tools for visualizing genomic context in different species. Figure 3.10 shows an example of a genomic context visualization created with Gosling-Meta. Here a phylogenetic tree is strongly linked with two metadata views. A phylogenetic tree shows the species' relatedness and a track heatmap encodes additional information using color. In this case, based on GeneSpy, the heatmap indicates which species are affected by a specific genetic event.

The goal of the reimplementation was to create a similar visualization to Figure 1 in the paper. The data for the genomic context visualization has been obtained from NCBI genome. The annotation files were used to extract the coordinates of the genes around the gene of interest. The absolute gene coordinates were transformed to local coordinates with the gene of interest at the center. A phylogenetic tree was calculated with Clustal Omega using the 16S rRNA sequences and transformed into JSON format. A further JSON file was introduced containing metadata on a specific gene event.

Figure 3.10 shows the reimplemented version and an excerpt of the corresponding JSON specifications. Two metadata views are attached to the Gosling visualization. A phylogenetic tree is specified in the Meta spec and a dummy track for placement is introduced in the Gosling spec. The dataId links the tree to the data of the genomic track, which is defined as a multi-row track in the Gosling spec. Internally the order of the leaves of the tree is applied to the rows of the Gosling track to ensure a consistent ordering. Therefore, the IDs used in the tree and heatmap data must correspond to the field used in the track (field: "Accession"). A metadata heatmap with a single column is placed between the tree and the genomic visualization. Details about the columns, including the domain and range of the values are specified in the Meta spec.

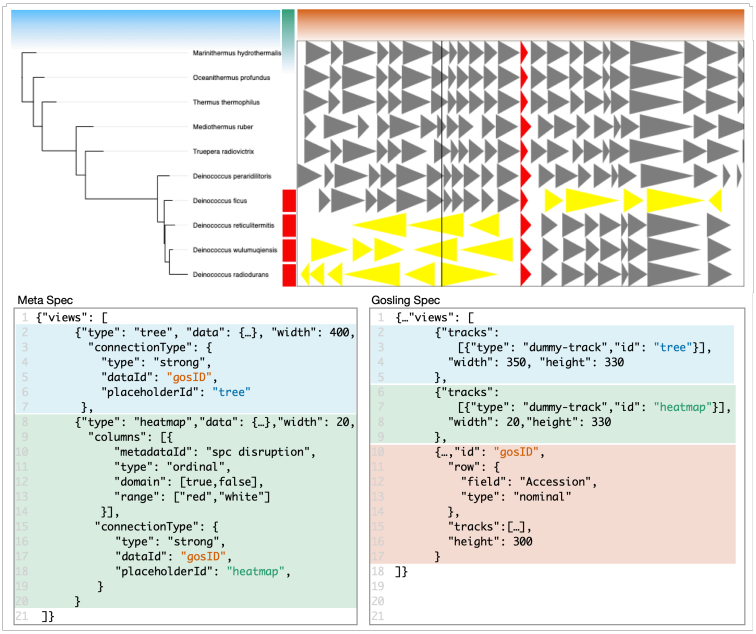


Figure 3.10: Example of a genomic context visualization created using `Gosling-Meta` demonstrating tightly aligned views. The visualization consists of a phylogenetic tree (blue), a heatmap (green), and a genomic visualization of a genomic neighborhood (orange). The `Meta spec` contains the specifications of the tree and the metadata heatmap. The `placeholderId` determines the location of the view by linking them to placeholder tracks (`dummy-track`) in the `Gosling` spec, while the `dataId` specifies the genomic data used for the axis alignment. For tightly aligned views a row track must be used in the `Gosling` spec.

Reimplementation: Metadata Table

`IslandViewer4` (reimplementation: Figure 3.11, original: Figure 3.2) has a genomic view and a weakly linked metadata table, showing feature-based metadata within a genomic window. The goal of the reimplementation was to create a similar visualization to the example on the `IslandViewer` website. The data was downloaded from the website and parsed into a format appropriate for Gosling visualization. Figure 3.11 shows the reimplemented version with an excerpt of the corresponding specifications. The table is defined in the `Meta spec` and linked to a placeholder track in the `Gosling spec`. In the `Meta spec` a `dataTransform` is used to merge two input data fields into one and the columns to be displayed are defined. The connection type links the data and the range displayed with the detailed `Gosling` view.

In this example, the field `jumpLinkage` is set to `false` in the `Meta spec`. When it is set to "true" a button at the end of each row of the table is introduced which can be used to navigate to the coordinates of the row in the `Gosling` visualization. This also makes it possible to display the entire data in the table and use it for navigation.

Furthermore, the example can be enhanced with other visualizations besides a table. Figure 3.12 shows alternative visualization specifications. While the `Gosling spec` remains the same, the `Meta spec` can be adapted to show a histogram (Figure 3.12a), a bar chart (Figure 3.12b), or a visualization specified using Vega-Lite (Figure 3.12c). A target column of the tabular gosling data has to be selected for visualization. Using data transforms the columns can be transformed before visualization. Figure 3.12a shows a `dataTransform` for calculating gene length using the start and end position.

3.3.6 Discussion

In the literature review only tools explicitly designed for prokaryotes were considered to limit the research scope and to fit the research focus on prokaryotes. While the biological questions are different for eukaryotes and prokaryotes, the types of visualizations are similar. The main differences are the smaller genome size and less complex genomic structure of prokaryotes, which leads to visualizations of

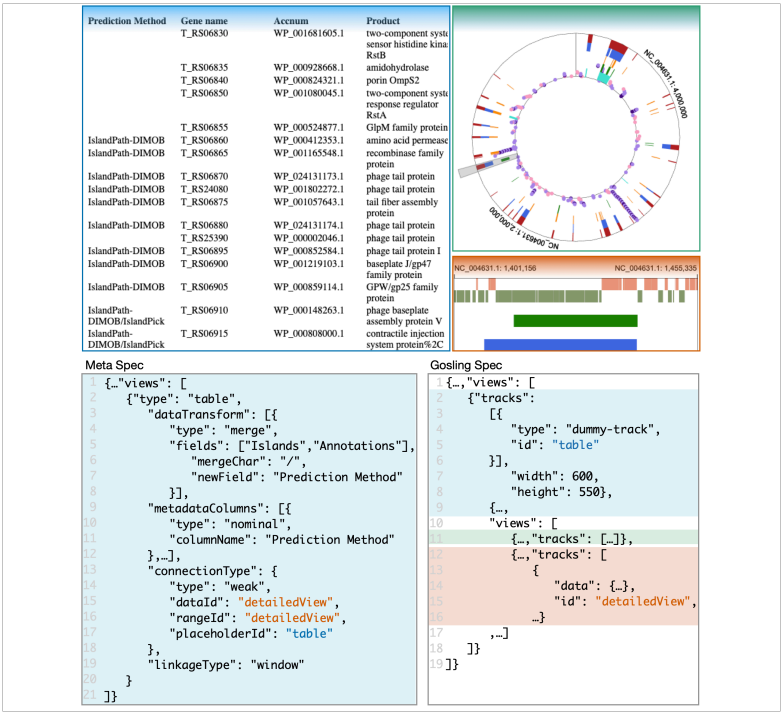


Figure 3.11: Example of a genomic island visualization created using `Gosling-Meta` demonstrating weakly aligned views. The visualization consists of a metadata table (blue), a visualization of the entire genome (green), and a detailed visualization of a genomic window selected with a brush (orange). The `Meta spec` contains the specification of the table. The `placeholderId` determines the location of the view by linking it to a placeholder track (`dummy-track`) in the `Gosling spec`, the `dataId` specifies the genomic data displayed in the table, and the `rangeId` specifies the genomic range displayed. Since the `rangeId` corresponds to the detailed view (orange) the metadata table only shows data within the same range. A `dataTransform` is applied to merge two columns of the input data and the metadata columns to be displayed are defined.

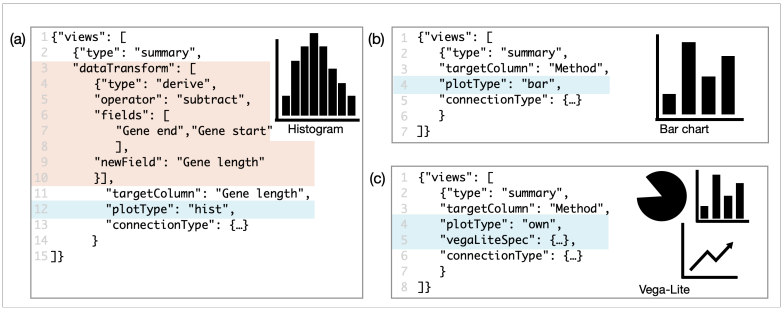


Figure 3.12: Weakly aligned plot options. (a) A histogram can be created if quantitative data are displayed in `Gosling` by setting the *plotType* to hist (blue). Here, a histogram of gene length is created. Using a *dataTransform* gene length is calculated using the fields specifying the start and end of genes (orange). (b) By setting `plotType` to bar (blue), a bar chart is created for categorical data. (c) If the *plotType* is set to "own" a `Vega-Lite` specification can be included and any plot can be created.

exon-intron structure and alternative splicing not being considered in the literature review. Moreover, track-based metadata has mostly been found in the form of trees. Metadata heatmaps are commonly used for large sample-based data sets where cohorts of patients are analyzed. While metadata on microbial samples certainly exists, it is not as abundant and complex as, for example, metadata on cancer patients.

The `Gosling` API has been extended to accommodate the injection of non-genomic visualizations. However, some limitations remain. `Gosling` uses `HiGlass` for its internal representation of tracks. While the two approaches share similar concepts, a `HiGlass` track is not equivalent to a `Gosling` track. The mapping between the two track types has been improved for `Gosling-Meta`, but the internal representation is not yet fully representative of the specification of tracks in the original `Gosling` specification, especially for overlaid tracks.

As an initial idea, I aimed at integrating non-genomic views directly into `Gosling`. This would allow placing non-genomic views directly using the nested view and track structure of `Gosling`. Moreover, it

would be a straightforward concept for users, as it accurately reflects the visualization structure. However, for the compartmentalization of functions and separation of code bases, placeholder tracks have been implemented. For direct integration, a joint specification could be implemented, while the existing concept could be used in the background to keep the code bases mostly separate.

In the current implementation of `Gosling`, there is no direct API access to the parsed data sets and data are only parsed if displayed in a track. This means that `Gosling-Meta` only has access to data displayed in genomic tracks within the currently displayed range. Therefore, if other data should be displayed in a `Gosling-Meta` view, invisible tracks containing the data must be created as a workaround. In future versions of `Gosling`, data sets associated with IDs parsed independently of the rendering can help alleviate this issue. Using the API, data sets could then be loaded directly into `Gosling-Meta`. Moreover, data in `Gosling` is currently specified for a track or a view and hierarchically passed down to tracks. Associating data with IDs could make them reusable between different views, which would shorten the specification and prevent parsing the same data multiple times.

Similarly, `Gosling-Meta` can only visualize data for a specific range if there is a `Gosling` track visualizing the specified range instead of obtaining the genomic range directly from a `Gosling` brush. The selection of genomic ranges using brushes within `Gosling` is not yet well reflected as an API functionality. Currently, two types of brushes exist for linear layout, one for linking the selected range with another `Gosling` view, and one for an external visualization, such as a `Gosling-Meta` view. Circular brushes can only be linked with internal `Gosling` visualizations. This can be exemplified for the `IslandViewer4` reimplementation example (subsection 3.3.5). The linear detailed view is needed for retrieving the data displayed in the table. It is currently impossible to remove the detailed view and only visualize the selected range in the table.

In conclusion, the work on `Gosling-Meta` has sparked many ideas on extending the `Gosling` API. While some ideas have already been implemented, such as placeholder tracks, others will be implemented in future releases. This is relevant for `Gosling-Meta` but also for

any project integrating `Gosling` visualizations programmatically. `Gosling-Meta` itself offers many visualizations to be combined with `Gosling` visualizations, which brings `Gosling` one step closer to a universal tool for building enhanced genomic visualizations.

3.4 Joint Conclusion

This chapter explored different levels of research on genomic visualization. First, the STAR contributes to shaping the field of genomic visualization by acting as a guide for the development of tools [8], [127], for the design of user studies evaluating genomic layouts [128], and for the direct translation of the visualization taxonomy into visualization grammar [25]. Second, `SeMa-Trap` represents visualization research with direct application in the biological domain. The tool development illustrates how systematic knowledge about the field of genomic visualization as well as the involvement of domain experts contribute to the development of domain-specific tools. Finally, `Gosling` and `Gosling-Meta` represent a middle ground of visualization research between categorizing the research space and developing approaches for end-users. The grammars are focused on translating the insights from the STAR into a tool for developers contributing to the quick development of genomic visualizations. It is easy to assess the impact of domain-specific visualization tools, for example when `SeMa-Trap` contributes to research on novel antibiotics. Yet, state-of-the-art reports and visualization grammars can be considered equally important, as they contribute to the evolution of the research area on a more foundational level.

Chapter 4

Visualizing Gene Function: GO-Compass

Multi-omics integration requires connecting the omics entities such as genes, transcripts, proteins, and metabolites across omics layers. One way of doing this is by analyzing the pathways or, more general, functions that they are involved in. The Gene Ontology provides a standardized vocabulary for describing function. `GO-Compass` uses the Gene Ontology to visualize the functional composition of lists of entities. Originally intended for visualizing lists of differentially expressed genes, the approach applies to any multi-omics entity translatable to Gene Ontology terms.

This chapter includes previously published work on `GO-Compass` presented at the EuroVis Conference 2023 [26]. For readability, funding information has been removed and figures originally included in the supplementary material have been included in the main text.

GO-Compass: Visual Navigation of Multiple Lists of GO terms

4.1 Abstract

Analysis pipelines in genomics, transcriptomics, and proteomics commonly produce lists of genes, e.g., differentially expressed genes. Often these lists overlap only partly or not at all and contain too many genes for manual comparison. However, using background knowledge, such as the functional annotations of the genes, the lists can be abstracted to functional terms. One approach is to run Gene Ontology (GO) enrichment analyses to determine over- and/or underrepresented functions for every list of genes. Due to the hierarchical structure of the Gene Ontology, lists of enriched

GO terms can contain many closely related terms, rendering the lists still long, redundant, and difficult to interpret for researchers.

In this paper, we present `GO-Compass` (Gene Ontology list comparison using Semantic Similarity), a visual analytics tool for the dispensability reduction and visual comparison of lists of GO terms. For dispensability reduction, we adapted the `REVIGO` algorithm, a summarization method based on the semantic similarity of GO terms, to perform hierarchical dispensability clustering on multiple lists. In an interactive dashboard, `GO-Compass` offers several visualizations for the comparison and improved interpretability of GO terms lists. The hierarchical dispensability clustering is visualized as a tree, where users can interactively filter out dispensable GO terms and create flat clusters by cutting the tree at a chosen dispensability. The flat clusters are visualized in animated treemaps and are compared using a correlation heatmap, UpSet plots, and bar charts.

With two use cases on published data sets from different omics domains, we demonstrate the general applicability and effectiveness of our approach. In the first use case, we show how the tool can be used to compare lists of differentially expressed genes from a transcriptomics pipeline and incorporate gene information into the analysis. In the second use case using genomics data, we show how `GO-Compass` facilitates the analysis of many hundreds of GO terms. For qualitative evaluation of the tool, we conducted feedback sessions with five domain experts and received positive comments. `GO-Compass` is part of the TueVis Visualization Server as a web application available at `https://go-compass-tuevis.cs.uni-tuebingen.de/`

4.2 Introduction

Lists of genes are one of the most common analysis results in bioinformatics. Often, multiple lists of genes are produced in large-scale molecular biology experiments, so-called “omics” studies. These

lists are interpreted and compared by bioinformaticians and biologists to gain insight into the structure, function, and dynamics of an organism.

In genomics, mutations or single nucleotide polymorphisms (SNPs) located in genes are of interest since they can alter the function of genes. For example, when comparing cancerous to non-cancerous cells, researchers aim to identify all genes containing mutations and study how these mutations may affect the function of the cell. Moreover, in microbiological studies, bacterial strains are compared which have specific properties, such as antibiotic resistance. Studying lists of genes showing SNPs between the strains can enhance insights into resistance mechanisms [129].

In transcriptomics and proteomics the expression of genes is studied, i.e., the intensity with which genes are transcribed into mRNA and translated into proteins. Usually, the expression of altered organisms (with induced mutations) and/or the expression at different conditions, such as timepoints or different environmental conditions, is studied. Multiple biological replicates are created for each condition, which allows calculating lists of significantly differentially expressed genes associated with fold changes between conditions. Moreover, genes can be clustered by their behavior across conditions resulting in a list of genes per cluster.

Comparing lists of genes is a non-trivial task. While sometimes genes can be compared individually, for example, if a gene is contained in multiple lists of differentially expressed genes, usually the lists overlap only partly or not at all. Therefore, gene lists are often compared in terms of their functional composition using underlying knowledge such as pathways or gene ontology terms associated with the genes [130].

The gene ontology (GO) is a graph-based, species-agnostic representation of information about the functions of genes, with respect to three domains: molecular function, biological process, and cellular component. It is organized in single GO terms encoding for specific functions. Every gene can be annotated with one or multiple GO terms. Each domain of the ontology is organized roughly hierarchically in a directed acyclic graph, which means that terms higher

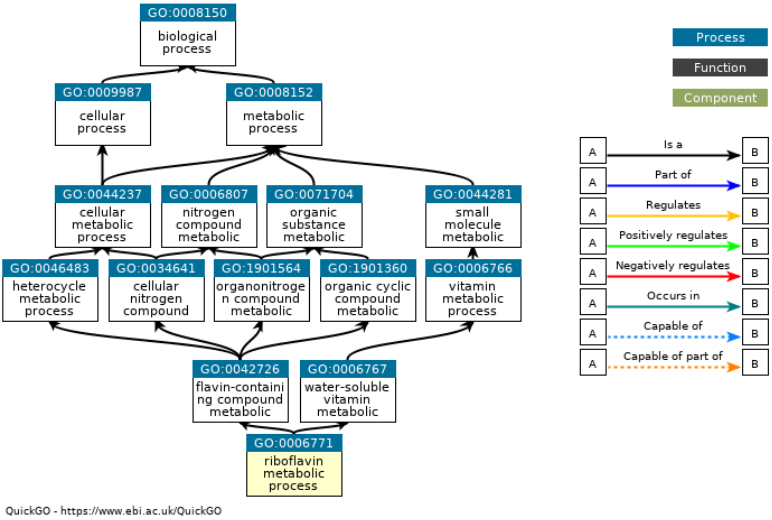


Figure 4.1: Example of GO term hierarchy for GO:0006771 created using QuickGO (`https://www.ebi.ac.uk/QuickGO/`)

up in the hierarchy are more general terms of function than terms further down the hierarchy (see Figure 4.1 for an example). This structure is both machine-readable and human-readable, which allows applying a range of methods that make use of the ontology and produce easily interpretable results. GO terms can be used to summarize the overall functions present in a list of genes, which is often done using gene ontology enrichment, a method to find statistically overrepresented GO terms [77], [131].

Not every gene is annotated with the same level of specificity. While some genes are annotated with very specific GO terms, others are annotated with very general terms. Therefore, the GO annotation of genes is usually propagated, which means that a gene is not only annotated with the specific GO term but also with all its parents to allow finding enriched GO terms even if not all the annotations are very specific. In practice, this means that a GO enrichment often results in long lists of GO terms containing many, possibly nearly identical terms. For example, if a very specific term, such as

riboflavin metabolic process (GO:0006771, Figure 4.1), is enriched, its parent terms including *water-soluble vitamin metabolic process* (GO:0006767), *vitamin metabolic process* (GO:0006766), and *small molecule metabolic process* (GO:0044281) might be present as well, but do not add much information.

While the structure of the GO graph causes this redundancy, it can also be used to combat it. The graph can be used to calculate the semantic similarity of GO terms which serves as a measurement to filter out highly similar terms [132]. One of the most simple similarity measures is the number of edges in the shortest path between two GO terms, but usually, more advanced methods are preferred. Our tool `GO-Compass` uses semantic similarity to hierarchically cluster GO terms. The tool lets users decide on the desired level of redundancy by visualizing the clustering result and allowing them to decide on a cutoff. The filtered GO term lists are then visualized in an interactive dashboard that contains comparative visualizations including an animated treemap, a correlation heatmap, an UpSet plot [100], and a comparative bar chart. We demonstrate how `GO-Compass` provides an efficient approach applicable in the omics domains by showing two use cases on published data sets. Moreover, in structured expert feedback sessions, we qualitatively evaluated our tool.

4.3 Related Work

For the downstream analysis and comparison of lists of GO terms various packages in R or Python have been developed. The Python package `goatools` implements functionalities such as statistical testing, summarizing gene lists, GO enrichment, and semantic similarity calculation [131]. The R package `Viseago` provides a comparative heatmap visualization of multiple sets of GO terms while clustering them by their semantic similarity [133]. However, the initial list of GO terms is not reduced, which can lead to very large heatmaps. `GOsummaries` is an R package that focuses on creating word clouds of GO terms [134]. It allows the processing of the data using filtering, clustering, and dimension reduction. While these packages are extremely powerful, since they can be easily

extended and combined with other packages, the requirement of programming skills limits the size of the user base. However, they can serve as the basis for the backends of other tools.

Furthermore, tools with a graphical interface have been developed, which can be accessed by a broader audience. `REVIGO` is a tool for reducing the redundancy in a list of GO terms and for visualizing the results [135]. The tool is based on a clustering algorithm, which ranks the GO terms by their dispensability based on their semantic similarity, their frequency, their p-values, and their relationship to each other in the GO graph. The visualizations consist of a treemap, a scatterplot, and a table. `REVIGO` is one of the most frequently used tools for reducing the redundancy and visualizing lists of GO-terms as reflected by its citation count. Moreover, the clustering result is used as an input for other tools, such as `CirGO`, a visualization tool that uses the `REVIGO` clustering results for visualization with sunburst charts instead of treemaps [136].

`BACA` is a tool for the comparative visualization of GO terms [137]. It does not only visualize GO terms, but it also includes information about gene expression. Each row in the visualization corresponds to a GO term and each column to a condition. For each condition and term, two circles are drawn corresponding to up- and downregulated genes. The size of the circles corresponds to the number of genes. The tool itself does not decrease the redundancy of lists of GO terms. Moreover, the number of genes might be a misleading value, as the size of sets of genes associated with different GO terms varies immensely due to the hierarchical structure of the GO graph.

4.4 Method

`GO-Compass` is a domain-specific tool for the comparison of lists of GO terms that arise from the analysis of large-scale omics experiments. The tool implements semantic similarity clustering of GO terms to reduce the redundancy in the lists. In a visualization dashboard, the clustering result is visualized and multiple visualizations are offered for the comparison of the reduced GO term lists.

GO-Compass makes use of several packages and methods introduced in Section 4.3. For GO enrichment and semantic similarity calculation, the goatools package is used. Moreover, GO-Compass includes an adapted re-implementation of the clustering algorithm of REVIGO that can handle multiple lists of GO terms. Similar to REVIGO, it implements a treemap for the visualization of the results of each list with an animated transition between treemaps of different lists. Gene information is incorporated using a glyph in the treemap. However, our visualization is not limited to gene lists from differential expression experiments, it can also be used for lists of GO terms and genes from other sources.

4.4.1 Data Input and Preprocessing

GO-Compass offers an internal enrichment method that can be used by submitting lists of genes. The genes can optionally be associated with numerical values (e.g., fold changes when calculating differential expression) that are used for visualization. Moreover, users can submit lists of GO terms with associated p-values arising from other enrichment methods. The GO term lists are arranged in a table where each column contains the p-values of one of the lists. In this case, the lists of genes are optional.

Furthermore, GO-Compass requires one or multiple background files containing the association of genes with GO terms for the enrichment and the clustering of GO terms. The lists of genes, GO terms, and backgrounds can originate from different species to enable interspecies comparisons or from different types of omics data (such as transcriptomic and proteomics) for multi-omics comparisons. GO-Compass propagates the backgrounds, which means that for every GO term, all parent GO terms are associated with its set of genes. Propagation can be disabled by the user if the lists have been previously propagated.

With a p-value filter, users can define their own significance level to filter out GO terms. Thus, only the GO terms are kept whose p-value is lower than the specified filter in at least one list. Moreover, a method for calculating the semantic similarity of GO terms can be selected, which is described in Section 4.4.2.

If lists of genes but no custom enrichment results have been submitted, `GO-Compass` performs GO enrichment with the Python package `goatools` [131] using the propagated background. For every list, the enrichment is calculated using Fisher's exact test. The resulting p-values are corrected using Benjamini-Hochberg's *False Discovery Rate* (FDR) [74].

4.4.2 Semantic Similarity

`GO-Compass` clusters GO terms based on their semantic similarity. There are various definitions of semantic similarity. The simplest measure is the shortest path between two GO terms A and B in the GO graph (Edge-distance measurement). A problem with this approach is that it assumes uniform distances in the graph. Since the GO graph is highly imbalanced, `GO-Compass` offers three other commonly used measures, the Resnik, Lin, and Wang semantic similarity [138]–[140]. Resnik semantic similarity is a node-based measure, defined as the information content (IC) of the lowest common subsumer of two GO terms, also called the *most informative common ancestor* (MICA) [138]. Let $S(A, B)$ denote the set of common ancestors of terms A and B and $p(t)$ the probability of term t. Then the Resnik similarity of A and B is defined as:

$$sim_{Resnik}(A, B) = IC(MICA) = \max_{t \in S(A,B)} (-\log(p(t))) \tag{4.1}$$

The Lin semantic similarity represents a normalized version of the Resnik similarity [139]:

$$sim_{Lin}(A, B) = \frac{2IC(MICA)}{IC(A) + IC(B)} \tag{4.2}$$

Wang semantic similarity is a graph-based method, which considers the contribution of all ancestors to the semantics of a term, with closer terms contributing more than more distant terms [140]. The

contribution of a term t to the semantics of term A is defined by the S-value:

$$S_A(t) = \begin{cases} 1 & \text{if } t = A \\ \max\{w_e \times S_A(t') \| t' \in \text{children of t}\} & \text{else} \end{cases} \quad (4.3)$$

w_e is the contribution factor of edge e linking t and t'. The semantic value is then the sum of all S-values of the set T_A which contains all ancestors of A:

$$SV(A) = \sum_{t \in T_A} S_A(t) \quad (4.4)$$

The Wang similarity between two terms A and B is then defined as:

$$sim_{Wang}(A, B) = \frac{\sum_{t \in T_A \cap T_B}(S_A(t) + S_B(t))}{SV(A) + SV(B)} \quad (4.5)$$

In this equation, the semantic contributions of the shared ancestors of A and B are summed up and divided by the total contribution of each term's ancestors.

`GO-Compass` uses `goatools` for the calculation of semantic similarity with these four measures [131]. At our request, the authors of `goatools` have also added Wang's method to their repertoire.

4.4.3 Algorithm

`GO-Compass` applies an adapted version of the `REVIGO` algorithm to perform hierarchical dispensability clustering separately for each GO domain. For every pair of GO terms, semantic similarity is computed and the pairs are sorted. Starting with the most similar pair A and B, a term is rejected, i.e., set as a child of the other term if one of several criteria, is fulfilled. The criteria are based on the terms' frequencies, their significance, and their relation in the graph. More concretely, the rejection criteria for term A are (applied in this order, and vice versa for B):

1. A has a frequency in the background data $> 5\%$ and the frequency of A is bigger than B.

2. The majority of p-values of A is less significant (difference >5% of the p-value range in the data).
3. A is the parent of B and contains more than 75% of its genes.
4. A is a child of B.
5. (Pseudo) random rejection: A term is rejected at random using the numbers of the GO ID corresponding to term A as a seed to guarantee that the result is consistent when the tool is run multiple times.

Criterion 3 guarantees that if two GO terms in a parent-child relationship are associated with a similar set of genes, the one that is less specific, i.e., the parent is rejected, while criterion 4 rejects the child if the sets of genes are different.

The semantic similarity at which a term is rejected corresponds to the term's dispensability. The term's uniqueness is calculated in the following way, where T corresponds to the set of all present GO terms:

$$uniqueness(A) = 1 - \frac{\sum_{t \in T} sim(A,t)}{|T|} \tag{4.6}$$

While `REVIGO` offers the Resnik, Lin, and two further IC-based methods, our adapted algorithm uses additionally and by default the Wang method, which takes the semantics of all ancestors of the GO terms into account to calculate their similarity. Moreover, the Wang method does not rely on the frequency of GO terms in the annotation of a species and is thus especially suited for comparisons using multiple different annotation backgrounds.

Furthermore, the p-value rejection criterion has been adapted to handle multiple p-values and thus to cluster terms from many lists of GO terms. Instead of rejecting the less significant p-value, the term for which the majority of p-values is less significant is rejected. The other rejection criteria have been implemented as described in the original `REVIGO` publication. Moreover, while the original algorithm stops clustering at a dispensability cutoff value provided by the user, `GO-Compass` always clusters all terms. The resulting structure is a tree with rejected terms as child nodes of accepted terms. The closer a term is to the root, the less dispensable it is.

4.5 Visualization

Based on our survey of related work we created multiple objectives for the development of GO-Compass:

1. Show all clustered GO terms to increase the explainability of the results. Users should see how their choice on a dispensability cutoff affects the final list of GO terms. Moreover, make terms that did not pass the filter accessible.
2. Provide visualizations that have a high acceptance and are commonly used in the field, like bar charts and REVIGO treemaps.
3. Summarize the overall similarity of the lists using, e.g., correlation values or the number of common enriched GO terms.
4. If available, also incorporate and visualize gene information.

Moreover, we refined the visualizations after testing the first version of the tool with domain experts in expert feedback sessions (see Section 4.7). In the following, we describe the implementation of our objectives in the current version of GO-Compass.

4.5.1 Dashboard Components

GO-Compass provides multiple visualizations organized in a dashboard for the comparison of the lists of GO terms based on the dispensability clustering. The dashboard consists of five components (Figure 4.2, original screenshot in Figure 4.3). Two components provide an overview of the data set by showing the clustering results (Figure 4.2A) or by summarizing the comparison of different GO term lists into single values (Figure 4.2C). The other components (Figure 4.2B and D) provide more detailed views of the individual GO term lists. All visualizations are interactive, provide linked highlighting, and are enhanced using tooltips displaying the full GO term names and further information. All GO terms together with their associated genes, p-values, dispensabilty, and uniqueness are collected in a table below the visualization components of the dashboard (see Figure 4.4 for an example).

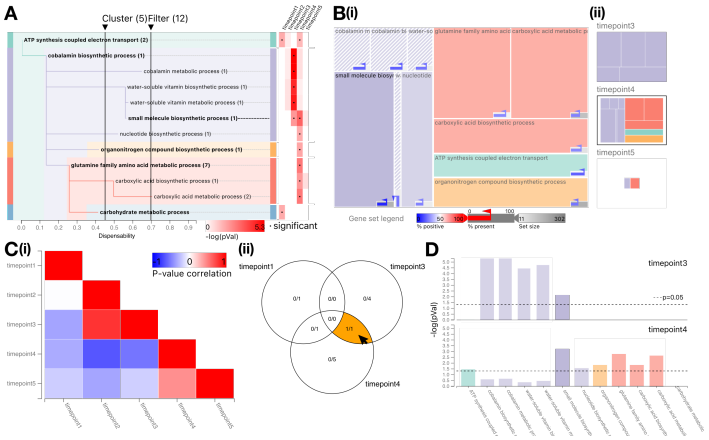


Figure 4.2: Overview of the dashboard components of `GO-Compass` using a data set from a time-series gene expression experiment [141]. (A) The `REVIGO` clustering results are visualized in an indented tree layout. Two sliders enable the interactive filtering of GO terms and the selection of a cutoff for the creation of flat clusters at chosen dispensability. The flat clusters are visualized using a set of categorical colors. The numbers next to the left and the right slider indicate the number of currently created flat clusters and the number of GO terms passing the filter cutoff. The number next to the GO terms indicates the number of direct descendants in the clustering that are currently filtered out. To visualize the p-values of each GO term, a heatmap is aligned to the tree on the right side. Guiding lines connect the GO terms in the tree with the heatmap. On the left side of the tree, an overview component is located which shows the current position in the tree is larger than the currently visible part. (B.i) A treemap shows the significance of GO terms at a selected list and encodes the flat clusters depending on the selected cutoff value. (B.ii) Using small multiples, different conditions can be selected. (C) A Venn diagram visualizes the intersection of significant GO terms across the different lists. The number of each subset represents the current selection (left) and the total size of each set (right). The subsets can be hovered (indicated by an orange highlight) for a more detailed comparison using bar charts showing the $-\log(p\text{-values})$ in the hovered lists. (D) The dotted lines represent the chosen p-value threshold.

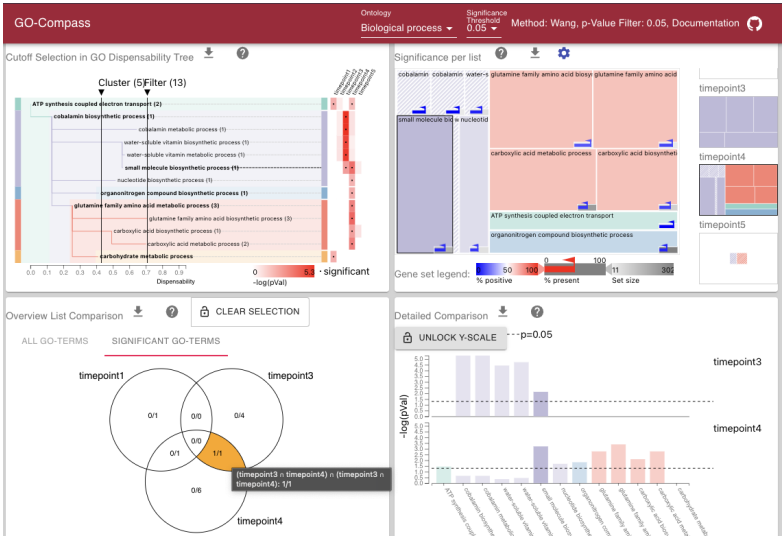


Figure 4.3: Original screenshot of the interface of `GO-compass`. It shows all visualization components as shown in Figure 1. For the view *Overview List Comparison*, only significant GO terms are used for the comparison, and hence a Venn diagram is shown.

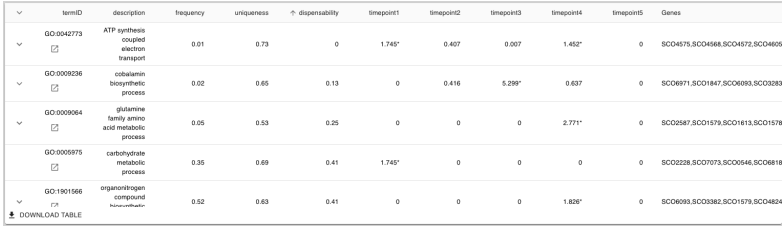


Figure 4.4: Screenshot of the table used in `GO-compass` to collect all visualized data. The columns show the ID of the GO term and its description. The next columns show the frequency, uniqueness, and dispensability for each term, as well as a detailed view of the p-values per list. The last column shows all genes present in the data set related to the GO term.

Using the controls menu of the dashboard, users can switch the GO domain displayed and define the significance threshold for the visualizations. Every visualization as well as the final filtered list of GO terms together with their p-values and dispensability can be exported.

Hierarchical Clustering Visualization and Cutoff Selection

To improve the explainability of the dispensability clustering results, created with the modified `REVIGO` algorithm (Objective 1), the clustering tree is visualized using an indented tree layout (Figure 4.2A). The nodes in the tree represent GO terms, positioned by their dispensability indicated on the x-axis. For interactive cutoff selection, two sliders in the tree can be used for selecting the data for the other visualizations. With the right slider, redundant terms can be filtered out. The left slider cuts the tree at a specific dispensability to produce flat clusters, which are colored using a set of categorical colors. These colors are used as the background color of the flat clusters in the tree, as well as in the other components of the dashboard. Next to each slider a number is shown indicating the number of flat clusters and the total number of GO terms currently visualized. Numbers next to the GO terms indicate the number of direct descendants of this term in the hierarchical clustering that are currently filtered out. By allowing an interactive selection of the cutoffs, users have full control over the number of clusters and GO terms that are visualized.

To facilitate the creation of flat clusters in the tree, child terms are always placed below parent terms in decreasing sort order by dispensability. Thereby, when a new flat cluster is created, all GO terms belonging to this cluster are placed in a block. Moreover, this layout prevents crossing edges and terms from changing positions when the filter slider is moved.

The p-values of the GO terms across different lists are visualized in a heatmap, which is aligned horizontally to the tree using guiding lines as a visual aid. The heatmap shows the negative logarithm of the p-values in the different lists using a color scale from white (high p-value) to red (low p-value). The p-values that pass the significance

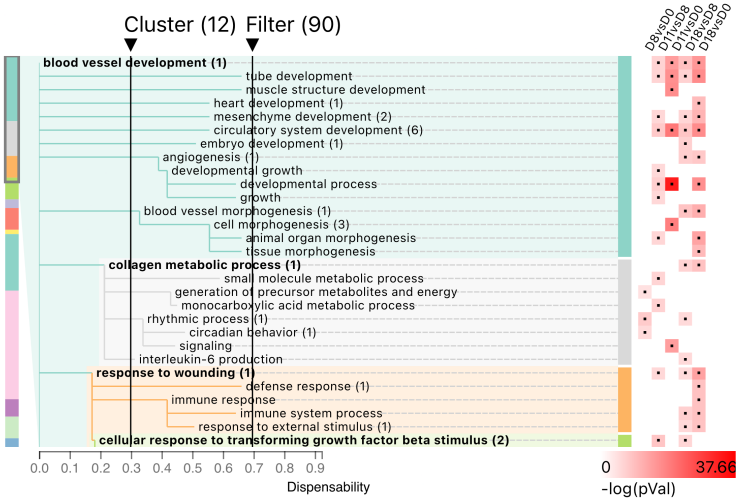


Figure 4.5: Example of a visualization in `GO-compass` for a large tree that is bigger than the available space. The summary visualization on the left of the tree shows the current position and size of the visible section of the tree. The colors of the summary visualization encode the colors of the flat clusters.

threshold defined by the user are indicated using a black dot in the center of the heatmap cell.

To deal with trees larger than the designated component, the tree and the aligned heatmap are vertically scrollable. A custom scroll bar that serves as a summary visualization on the left side shows the current position and size of the visible section of the full tree relative to the currently created flat clusters (Figure 4.5).

Treemaps

For list-centered overviews, the significance of GO terms at each condition is visualized using treemaps, similar to *REVIGO* treemaps (Objective 2, Figure 4.2B). The size of a rectangle corresponds to the negative logarithm of the p-value. Rectangles representing GO

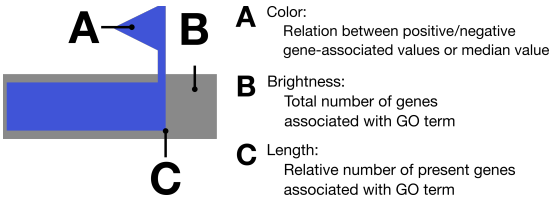


Figure 4.6: Glyph for the visualization of gene information. (A) The color of the bar either encodes the median value in a set of genes or the proportion of positive/negative values on a color scale from blue to white to red, where blue and red shades correspond to the majority of values being negative and positive, respectively. A white shade encodes an equal number of positive and negative values. (B) The brightness of the background in the glyph corresponds to the set size of the GO term, where smaller sets receive darker fills than bigger sets. (C) The length of the bar encodes for the relative number of present genes associated with the GO term.

terms with significant p-values receive a full color fill, and non-significant GO terms are indicated with a striped fill. One list is selected to be shown as the main treemap (Figure 4.2B.i), while the others are visualized in small multiples in a scrollable container (Figure 4.2B.ii). By clicking on a small multiple, the corresponding list is selected and triggers an animation where the rectangles dynamically change their size. This helps find especially prominent changes between lists. For smooth transitions, the treemap is implemented using a modified version of the squarify algorithm [142], where nodes only change size but not positions when the animation is triggered.

A glyph is used to additionally show gene information if available (Figure 4.6, Objective 4). The glyph consists of a bar showing the proportion of genes associated with a GO term that are present in the corresponding genes. The color shade of the bar is used to encode additional numerical information if the genes are associated with numbers (e.g., fold changes in differential expression). Users can choose to either show the median value or the proportions of negative to positive values on a color scale from blue to red. A small flag-like attachment on the right end of the bar ensures that the color

is still recognizable even if the bar is short. If no information about fold changes is available, the bar is colored black. Furthermore, the background of the glyph encodes the gene set size of a GO term using a grayscale color scale, where dark values encode for larger sets than bright values. If no gene information has been uploaded, only the gene set size is shown, which is always available since it is calculated using the uploaded background lists.

When hovering over a rectangle in the treemap the p-values of the corresponding GO term in the different lists are displayed in a tooltip with bar charts or line charts. Moreover, the full GO-term name, as well as the gene information is shown in the tooltip.

Summary Visualizations and Bar Charts

In order to allow a quick comparison of the lists (Objective 3), two summary visualizations show the similarity of GO terms between the lists (Figure 4.2C). The user can choose between visualizing the correlation of the p-values of the GO terms (Figure 4.2C.i) or the intersecting sets of significant GO terms across the lists (Figure 4.2C.ii). A correlation heatmap visualizes Pearson's correlation of the p-values of GO terms between the lists regardless of the significance threshold (as seen in the use case in Figure 4.7B) using a blue (negative correlation) to red (positive correlation) color scale. The set summary visualization shows the overlap of significantly enriched GO terms between the conditions. For less than four lists, a Venn diagram is used (Figure 4.2C.ii). For more than three lists an UpSet plot is visualized instead (as shown in the use case in Figure 4.7D) [100]. UpSet plots show set intersections as a matrix, where rows correspond to the sets and columns to the intersections.

For detailed comparison of subsets of lists, cells in the correlation heatmap or intersections of the Venn diagram or UpSet plot can be hovered and the GO terms can be compared in vertically juxtaposed bar charts (Figure 4.2D), where each bar chart corresponds to one of the hovered lists. All GO terms present in the hierarchy after filtering are shown on the horizontal axis.

4.5.2 Implementation

GO-Compass is implemented as a client-server application using Python for the enrichment calculation and dispensability clustering with goatools in the backend and JavaScript for the visualization in the front end. The Python package Flask [143] is used for client-server communication. The frontend is implemented using a combination of React [144] for the component structure of the application, Mobx [145] for storing the application's state, and D3 [146] for the visualizations, i.e., scales, animations, and the treemap layout.

4.6 Use Cases

To demonstrate the general applicability of GO-Compass in the omics field, we present two use cases. The first use case shows how GO-Compass allows the exploration of many GO term lists from a transcriptomics experiment and also includes further important gene information. In the second use case, we present the results of an enrichment analysis of genomics data and demonstrate how GO-Compass facilitates the analysis of a few hundred of GO terms.

4.6.1 Use Case 1: Functional Enrichment of Antibiotic Response in the Mouse Transcriptome

In the study by Lavelle et al. [147], the authors addressed the question how antibiotics change the transcriptome response across a time window of 18 days in two mice. For this, they sampled RNA at four different time points (Day 0, 8, 11, and 18), analyzed the expression levels using microarrays, and computed differentially expressed genes between each time point. Of the six pairwise comparisons, one did not return significant differential expression (day 18 compared to day 11, in short D18vsD11). To acquire a functional overview of the differences, the authors ran an enrichment analysis using goana [148] on the five lists of genes. Independently for each comparison, they extracted the 100 most significant GO terms and reduced the data sets to those terms with a dispensability lower than 0.4 by using REVIGO for each of the lists. For visualization of the results they

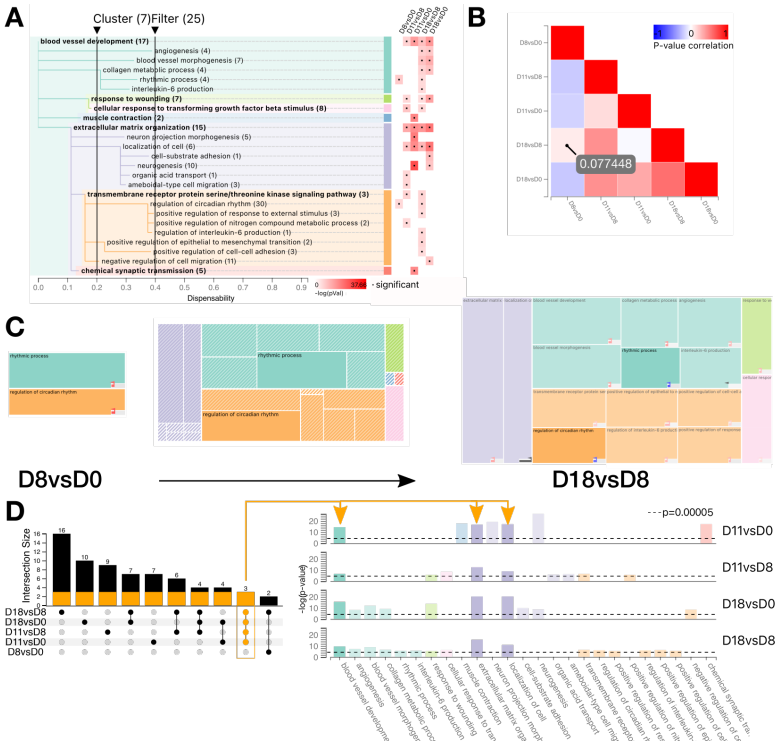


Figure 4.7: Use case for `GO-Compass` with the transcriptome data from the study by Lavelle et al. (2019). All subfigures visualize terms from the ontology domain *Biological process*. (A) Visualization of the clustering results. The clusters were cut at a dispensability level of 0.2 using the left slider. Terms with a dispensability lower than 0.4 have been filtered out using the right slider. (B) Correlation heatmap of all p-values of the terms contained in the five lists. The list comparing day 8 with day 0 (D8vsD0) shows an anticorrelation of GO terms with all other lists, except for a minimal positive correlation with the list D18vsD8. (C) Treemap visualizations for the conditions D8vsD0 and D18vsD8. The transition animation (middle treemap) helps to follow the changes between the two treemaps. The glyph shows the median $\log_2$-fold change of the genes linked to each GO term. (D) UpSet plot comparing the specific terms shared across conditions. Three GO terms are found significant in all four lists. The juxtaposed bar charts containing all terms of these four lists are shown, and the bars for the common terms are highlighted.

created a tailored visualization using custom R scripts. Figure 3C in the original article visualizes the results using bar charts encoding for the negative logarithm of the p-value [147].

We reproduced the results with `GO-Compass` to show how it is able to provide the same information and furthermore include exploratory as well as other informative features. The preprocessing was reproduced from the data and scripts provided by Lavelle et al., resulting in five lists of differentially expressed genes for each comparison (referred to as D8vsD0, etc. in the following) and the significant GO terms identified by `goana`. To make the results comparable, we also reduced the data set of each comparison to the 100 most significant terms as done by the authors. The five lists of genes with their $\log_2$-fold changes, the results of the GO enrichment, and the used GO background were loaded into `GO-Compass` for the dispensability computation and visualization. Wang's similarity measurement and a p-value filtering of 5×10^{-5} were selected as further parameters.

We first focus on those terms related to the domain *Biological process*. For this domain, `GO-Compass` returns a visualization with 179 GO terms, which can be reduced to 25 terms by filtering terms with a dispensability higher than 0.4 (Figure 4.7A), as done in the original analysis. To gain more insight into the different GO terms, the cluster cutoff of `GO-Compass` was set to 0.2, returning seven flat clusters (Figure 4.7). The visualization of many aspects of the lists within the graphical interface allows simultaneous exploration. For example, the heatmap aligned next to the tree shows that all but the first list of GO terms contain terms of the cluster rooted at *extracellular matrix organization*. This dissimilarity can be confirmed by looking at the correlation heatmap (Figure 4.7B) that shows how D8vsD0 is negatively correlated with all but one comparison at this dispensability level (D18vsD8, Pearson correlation $\rho \approx 0.08$).

By using the treemaps, users are able to compare the flat clusters between these two lists and account for further data, such as the $\log_2$-fold change of the genes. The animated transition facilitates the finding of the common terms between the lists and tracking major changes (Figure 4.7C). The visualization of the median shows that the expression of genes corresponding to the lists is not similar.

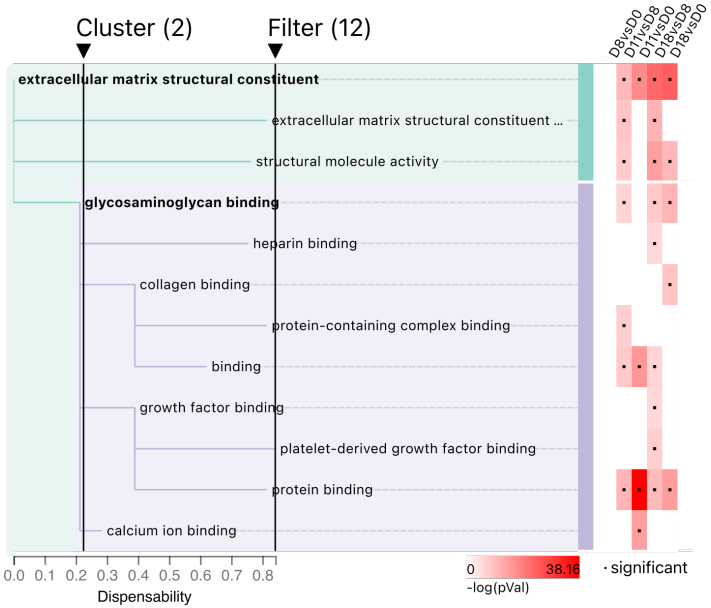


Figure 4.8: Tree showing the clustering results for all terms related to the ontology domain *Molecular function* for the study by Lavelle et al. (2019). The cluster cutoff was defined at a dispensability level close to 0.2 to visualize a cluster rooted at *glycosaminoglycan binding* (purple cluster).

While the genes related to the terms *rhythmic process* and *regulation of circadian rhythm* in D8vsD0 are upregulated, the genes in D18vsD8 are strongly downregulated (see glyphs in Figure 4.7C).

The UpSet plot can be used to compare the terms that are significant in more than one list (Figure 4.7D). For example, the GO terms *extracellular matrix organization*, *blood vessel development* and *localization of cell* are found significant in four lists and their corresponding p-values can be easily compared using the bar charts. Interestingly, in the dispensability reduction of the original analysis, the term *extracellular matrix organization* appears only in three of the five lists. However, this term is actually significant in four of the lists, which is evident in the analysis with `GO-Compass`.

Lavelle et al. also analyzed significant GO terms for the ontology domain *Molecular function*, for which they identified ten different GO terms, all related to *binding*. With the visualization of `GO-Compass`, the user is able to visualize the hierarchy of the clusters and identify the most important terms regarding their dispensability (Figure 4.8). In this case, by setting a cluster cutoff at 0.2 a meaningful flat cluster is created containing all terms related to *binding* rooted at *glycosaminoglycan binding*. Moreover, by visualizing the whole clustering and interactively selecting the filter cutoff, the user can decide whether general terms such as *binding* should be filtered out to reduce the redundancy of the results for further analysis. This cutoff is independent of the filter cutoffs selected for the other GO domains, while in the original analysis with `REVIGO` a single cutoff value for all domains had to be chosen.

4.6.2 Use Case 2: Genomic Variability in the Syphilis Agent, Treponema pallidum

We demonstrate the wide applicability of `GO-Compass` for studying the genetic variability of *Treponema pallidum*, the bacterium responsible for syphilis [149]. For this, the data published by Pla-Díaz et al. [150] was analyzed using the tool `Evidente`[13] to identify enriched GO terms for genes associated with SNPs in phylogenetic clades. A clade is a set of organisms that share a common ancestor in a phylogenetic tree. The data of Pla-Díaz et al. consists of genomic samples from four main strains of *T. pallidum*: ten samples from the strain *Nichols* and 58 samples from the strain *SS14* (both belong to subspecies *T. pallidum pallidum, TPA*), seven samples from subspecies *pertenue* (TPE) and one sample from subspecies *endemicum* (TEN). All samples were compared to one *Nichols* reference genome to identify single-nucleotide polymorphisms (SNPs). Using `Evidente` we identified GO terms that are enriched for genes affected by mutations within one clade compared to the rest of the phylogenetic tree. A GO enrichment was calculated for four main phylogenetic clades, producing a list of significantly enriched GO terms per clade (p-value < 0.05). All non-reference clades showed enriched GO terms: 61 within the SS14 clade, 70 within TPE and

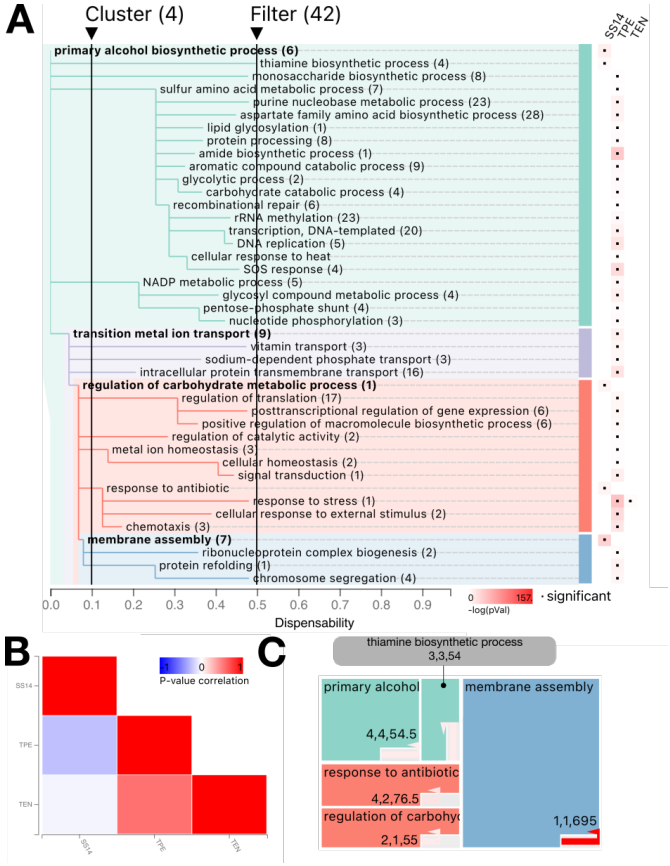


Figure 4.9: Use case for GO-Compass resulting from the GO enrichment of phylogenetic clades of *Treponema pallidum*. All subfigures visualize terms from the ontology domain *Biological process* with a dispensability lower than 0.5. (A) Visualization of the clustering results. The cluster cutoff was set at a dispensability level of 0.1. (B) Correlation heatmap of all p-values of the terms contained in the three clades. (C) Treemap visualization for the SS14 clade. The numbers above or beside the gene information glyph encode the total number of genes associated with the GO term, the genes associated with the GO term containing SNPs, and the median number of SNPs found within these genes. The same values are encoded in the gene information glyph using the background color, the length of the bar and the color of the bar, respectively.

514 within the TEN clade. Since `Evidente` does not contain features for direct comparison of GO terms, these three lists, containing a total of 576 unique significant GO terms, were loaded into `GO-Compass`. Furthermore, for each clade, a list containing the number of SNPs per gene across all strains and positions was uploaded as further gene information. Wang's semantic similarity measure and p-value < 0.05 were used as parameters for the redundancy calculation.

All three GO domains (biological process, molecular function, and cellular component) show high correlation and GO term similarity across the TPE and TEN lists, while the SS14 clade shows distinct GO terms over all three domains. We further reduced the data by filtering out GO terms with a dispensability larger than 0.5. In the case of the domain *Biological process*, the data set of 306 GO terms was reduced to 41 terms which still reflects the similarities between TPE and TEN as well as their differences with SS14 (Figure 4.9A/B). Interestingly, the GO terms specific for the SS14 clade include low dispensability terms such as *membrane assembly* and *response to antibiotic* (Figure 4.9A). The low dispensability and the position of the terms in the tree indicate that they are specific to the SS14 clade and are only very distantly related to terms enriched in TPE and TEN. This is in agreement with the information presented by Pla-Díaz et al., where they discuss the changes in the membrane of SS14 strains and the antibiotic-resistant potential across SS14 samples when compared to strains from the Nichols clade. With the glyph visualization in the treemap, we were able to identify that a single gene is linked to the term *membrane assembly*. The red color of the bar indicates that there is a high number of SNPs (695) associated with this gene across all samples and positions of the gene (Figure 4.9C). This specific gene (*TPANIC_RS01590*) can be identified using the tabular view for further analysis, for example, to study the effect of the SNPs on the translated protein.

4.7 Qualitative Evaluation

For qualitative evaluation of our tool, we performed semi-structured interviews with five domain experts including two bioinformaticians,

Table 4.1: Tasks for the empirical case study

Task 1	Simplify the visualization/reduce the size of the tree.
Task 2	Which lists are the most similar/least similar ones?
Task 3	Which GO terms are the three most significant GO terms shared between the two most similar lists?
Task 4	Summarize the biological processes happening at D11vsD0.
Task 5	For the GO term *regulation of transmembrane transport* is the majority of the genes upregulated or downregulated in list D11vsD0? Which fraction of genes is expressed? Is the set of genes associated with this GO-term higher or lower than *regulation of biological quality*?

two biologists, and a clinical doctor who is one of the authors of the original publication discussed in the first use case (see Section 4.6.1). None of the experts was involved in the development of the tool or had used our tool before. The interviews were conducted in an online video session and took around 1h each. First, the experts were interviewed about their experiences using GO terms. Depending on their level of expertise the concept of Gene Ontology was recapitulated and the central ideas of the tool were introduced. Apart from the dispensability clustering tree, none of the visualizations were explained to get an unbiased view of the tool's intuitiveness. After the session, the experts were asked to fill out a short survey that includes a selected subset of the system usability scale questionnaire (see Table 4.1 for the questions) [151].

For the testing, the data from the first use case was used. During the session, the experts were presented with five tasks, each related to a different visualization component (Table 4.1). The first task (T1) was to reduce the size of the dispensability tree and simplify the visualization. In the second task (T2) users were asked which lists are the most/least similar ones. For the third task (T3) they were asked which three GO terms are the most significant ones shared between the two most similar lists. The fourth task (T4) was about gaining an overview of a single list and the experts were asked to

summarize the biological processes happening in the list D11vsD0. The last task (T5) was made up of subtasks for comparing the gene sets between two GO terms in the treemap based on the underlying gene information.

4.7.1 Expert Feedback

All participants were able to successfully reduce the size of the tree (T1) by using the filter slider. Two participants were initially confused by the cluster slider but were able to figure out its function when being told to observe the effect of moving the slider in the other visualizations. Moreover, all participants were able to compare lists using the correlation heatmap (T2). While they were not able to tell which values are correlated (p-values), they had the intuition that similar lists were indicated by a red color (see Figure 4.2C,i). All participants were able to compare the lists in detail using the bar chart (T3) and found the linkage between the correlation heatmap and the bar charts useful.

For the fourth task, all users were able to navigate to the list in question in the treemap visualization and could summarize the biological processes. Similar to the second task, users were not able to explicitly tell what is encoded, but still were able to solve the task. The glyph visualization (T5) received mixed feedback from the users. It is important to note, that the version tested did not include a legend for the gene set glyph. Two users wrongly assumed that the blue-to-red color scale of the glyph is related to the correlation displayed in the correlation heatmap. Moreover, initially, two users could not tell that the background of the glyph was colored using different shades of gray. In the tested version big gene sets received brighter fills than small gene sets, which was considered unintuitive by all users. We, therefore, decided to reverse the color encoding of the gene set sizes for the final published version of `GO-Compass`. Eventually, all users were able to solve the subtasks and considered the glyph useful after the explanation.

Overall, the tool received positive feedback from all experts. One expert noted that although she could not tell why it was very intuitive for her to know where to click and hover. While the visualization

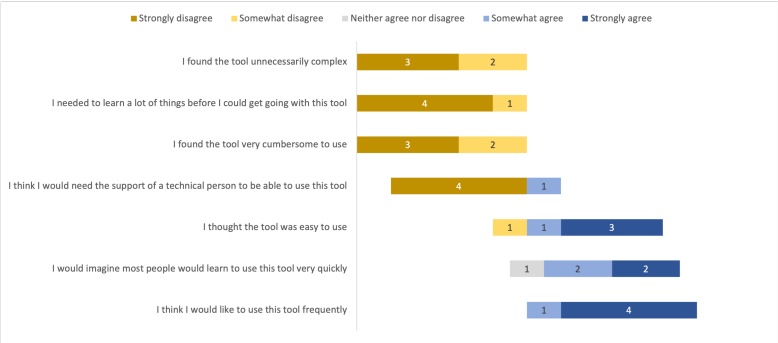


Figure 4.10: Survey results for empirical case study. Overall, the visualization received positive feedback.

dashboard was first considered slightly overwhelming by some users, all found it useful and consider using it for their own research or for teaching. One expert noted, that the tool obviously shows a lot of information. However, they did not consider it to be a flaw as the underlying data contributes a lot to the complexity and the tool already reduces the mental load for analyzing the data. One expert noted that he "can't think of anything that beats this" when being asked for an overall evaluation of the tool.

From the feedback session, we deducted several action items and features for implementing the tool that we then included in the current version. Many of these features relate to improving the labeling and legends of the visualizations as well as solving minor usability issues. Legends were included for the glyph and a label now indicates that the correlation heatmap shows the correlation of p-values. Moreover, the scrolling behavior of the dispensability tree was improved by making the overview visualization draggable. Furthermore, a user suggested a more detailed visualization of gene information as well as a simplified version of the dispensability tree which we consider enhancements for future versions of the tool. The answers collected from the survey after the interview supported the positive impression we had during the interviews (Figure 4.10). While the tool was considered complex, all users found it useful and would like to use it frequently for their own research.

4.8 Discussion

We presented GO-Compass, an application with a unique set of features enabling the interactive visual comparison of lists of gene ontology terms. The tool extends the REVIGO algorithm and the treemap visualization for the analysis of multiple lists of GO terms and has a wide field of applicability for lists created in different omics studies due to the simple input formats it requires. With its tree visualization, it increases the explainability of the underlying hierarchical clustering and gives users full control over the desired level of dispensability in the lists of GO terms. Furthermore, it effectively combines well-known visualization techniques, for the direct comparison of sets such as UpSet plots and correlation heatmaps with detailed-focused visualizations like bar charts and treemaps for a comprehensive analysis. Moreover, with the glyph in the treemaps gene information is included, if available.

GO-Compass is based on the structure of the gene ontology, semantic similarity calculations, and on GO enrichment. Since the GO graph is created using expert knowledge, which differs for the different sub-domains of the graph, it lacks uniform density and uniform relatedness of parent-child relationships. The edges of the GO graph are not weighted according to relatedness, therefore it is not possible to take this factor into account for semantic similarity calculation. GO-Compass counteracts these factors by allowing interactive selection of the filter cutoff and therefore gives users the opportunity to decide on a desired level of redundancy with computational support. Furthermore, different dispensability cutoffs can be chosen for each gene ontology domain. Thus, with our interactive dispensability filtering we compensate for the bias that all GO terms are treated the same regardless of their relatedness in the GO graph. When reducing the dispensability in lists separately, for example using REVIGO, it can happen that a term is filtered out in some of the lists despite being significant in all of them. Our simultaneous analysis of all lists avoids losing information, as seen with the term *extracellular matrix organization* in the first use case, which was filtered out in one of the lists in the original analysis.

GO-Compass clusters the complete lists of GO terms instead of stopping at a fixed dispensability cutoff. This gives the users insights into the underlying algorithm and increases their control over the desired level of redundancy. This can be seen especially prominently in the first use case, where dispensable terms could easily be identified and filtered out using the visualization. The indented layout of the tree is designed to facilitate the process of filtering and the creation of flat clusters, as it ensures that all children of a GO term in the hierarchical clustering are organized in a block without the need of rearranging the tree after moving the sliders.

GO-Compass implements treemaps for visualizing the functional composition of a single list. An animation is triggered to show the changes when the condition is switched. While it is difficult to compare the areas of rectangles of different sizes and follow multiple concurrent animations, the most drastic changes are underlined using the animation. A more detailed comparison can be done using the bar chart appearing in a tooltip when hovering over a GO term. Moreover, with the designed glyph any type of numerical information associated with genes can be included in the analysis. Moreover, by including linked summarizing charts (the correlation heatmap and the UpSet plot) and detailed views (treemap and juxtaposed bar charts), users can access different levels of information of the data sets. This helps to compare the lists using many relevant aspects for the domain, such as further gene information or shared GO terms.

In our use cases, we demonstrated the applicability of GO-Compass for two different omics data types. In our first use case, we presented a use case for functional analysis in transcriptomics studies, which is one of the most common use cases where GO enrichment is applied. In the second use case, we show how the tool can be applied to genomics data and handle data up to a few hundred GO terms. However, GO-Compass can be applied in many other contexts, such as the integration of lists originating from different omics layers in multi-omics experiments, as well as lists originating from different species.

GO-Compass compares lists of genes regarding their functional composition using GO terms. However, also other annotation systems,

such as KEGG Pathways or protein families, are frequently used by researchers for functional analysis. Compared to these annotations, the advantage of GO terms is the inherent structure of the GO graph, which allows the simple calculation of semantic similarity to serve as the basis for dispensability clustering. Nevertheless, we plan to extend our tool to be able to deal with other types of enrichments using custom algorithms for dispensability calculation.

To conclude, as omics experiments increasingly consist of the comparison of many conditions, semantic-similarity-based reduction for generating visualizations of multiple GO term lists has become more and more popular and useful. With `GO-Compass` we provide an easy-to-use visualization tool for this task with the ultimate aim to help researchers interpret the biology of the systems studied.

Acknowledgments

We thank Dr. Aonghus Lavelle for providing the raw data for the reproduction of the first use case as well as for providing his expert opinion in the expert feedback session. Moreover, we thank Nadine Ziemert, Bernhard Krismer, Julian Fratte, and Evi Stegmann for taking part in the expert feedback sessions. Furthermore, we thank Haibao Tang and DV Klopfenstein for the implementation of Wang's semantic similarity method in `goatools`.

Chapter 5

Visualizing Cluster Patterns: OmicsTIDE

Mutli-omics data can be integrated using joint clustering in an *early integration* approach. This is especially useful for transcriptomics and proteomics data as they are similar from a data perspective. `OmicsTIDE` leverages this property by performing joint clustering of transcriptomics and proteomics abundance data with shared conditions and visualizes the result.

This chapter includes previously published work on `OmicsTIDE` [27]. For readability, funding and contact information on the corresponding author have been removed. Figures and tables originally included in the supplementary material have been included in the main text.

OmicsTIDE: Interactive Exploration of Trends in Multi-Omics Data

5.1 Abstract

Motivation: The increasing amount of data produced by omics technologies has enabled researchers to study phenomena across multiple omics layers. Besides data-driven analysis strategies, interactive visualization tools have been developed for a more transparent analysis. However, most state-of-the-art tools do not reconstruct the impact of a single omics layer on the integration result.
Results: We developed a data classification scheme focusing on different aspects of multi-omics data sets for a systemic understanding. Based on this classification we developed the Omics Trend-comparing Interactive Data Explorer (`OmicsTIDE`), an interactive visualization tool for the comparison of gene-based

quantitative omics data. The tool consists of a computational part that clusters omics data sets to determine trends and an interactive visualization. The trends are visualized as profile plots and are connected by a Sankey diagram that allows for an interactive pairwise trend comparison to discover concordant and discordant trends. Moreover, large-scale omics data sets are broken down into small subsets that can be analyzed functionally using Gene Ontology enrichment within few analysis steps. We demonstrate the interactive analysis using `OmicsTIDE` with two case studies focusing on different experimental designs.
Availability: `OmicsTIDE` is a web tool available via http://omicstide-tuevis.cs.uni-tuebingen.de/

5.2 Introduction

With the advent of high-throughput technologies, it has become affordable to comprehensively study all entities in one omics layer of a sample, e.g. all genes, transcripts, proteins, or metabolites. While studying single omics layers already requires sophisticated method, analyzing multiple omics layers across several experimental conditions adds a whole new level of complexity. Therefore, the demand for methods that integrate and visualize multiple omics data sets has been steadily increasing over the past decades.

While a data-driven integration can derive interesting relations between different omics layers, it often is perceived as a black box. For instance, the impact of single genes or groups of genes on the integration is not always evident. To overcome this limitation of purely data-driven methods, different approaches have been developed [21], [152]. Many of these approaches reduce the complexity of the data sets by classifying single genes into different categories based on their "behaviour". For example, in a data set that deals with two conditions, a gene could be classified as being *up-regulated* in one condition with respect to the other condition. The situation becomes more complex when more than two conditions are observed. This requires the application of clustering methods to obtain representative *trends* for sets of genes. Here, we define a

trend in omics abundance data as a set of omics-entities that follow a distinct trajectory across at least two conditions.

In order to provide a tool that overcomes the current limitations in the omics visualization field, we first devised a general classification system for omics data that categorizes the data to be analyzed and compared with respect to their data type as well as experimental design. A detailed categorization preceding an integrated omics analysis will help to choose a suitable analysis approach. This classification builds the framework for the *Omics Trend-comparing Interactive Data Explorer* (`OmicsTIDE`), a tool that creates a connection between the single genes and the trends derived from gene-based quantitative multi-omics data sets including, for example, transcriptomic and proteomic data.

The comparison of trends found in two omics layers is the central concept of `OmicsTIDE`, which visualizes trends as profile plots, also known as parallel coordinate plots [153], and compares trends between two data sets using a Sankey diagram [154]. `OmicsTIDE` aims to identify the same trends in both data sets, therefore, genes are further grouped into whether they follow concordant (i.e., the same) or discordant (i.e., different) trends. Moreover, the tool breaks down large-scale data sets into small subsets within a few steps based on selecting groups of genes in the Sankey diagram. These subsets can be functionally analyzed using Gene Ontology enrichment analysis. By allowing several pairwise comparisons within a single analysis, `OmicsTIDE` combines insights from different pairwise comparisons into one large analysis. We demonstrate the effectiveness of `OmicsTIDE` in two case studies with different experimental designs.

5.3 Related Work

For this paper, we define a multi-omics tool as a tool that integrates and visualizes data of two or more omics layers concurrently. In this related work section, we therefore focus on tools that analyze different omics data in a combined instead of a separated or sequential manner.

The most straightforward way of visualizing multi-omics data is mapping them directly to a genome sequence or a pathway. Any kind of omics data that can be mapped to a genome sequence can be represented in genome coordinate-based visualizations such as genome browsers [24]. With tracks stacked upon each other, various omics layers can be displayed simultaneously. Similarly, omics data can be mapped to a pathway ID in a node-link diagram, where genes, proteins, and metabolites can be shown simultaneously [20]. While genome browsers and pathway maps intuitively visualize multi-omics data, they usually show only a small window of the genome or a single pathway of interest and are limited to a small number of conditions that can be displayed simultaneously.

Moreover, various computational methods have been developed for the integration of multi-omics data, as reviewed by Bersanelli, Mosca, Remondini, *et al.* [155] and Huang, Chaudhary, and Garmire [156]. Often omics data are clustered using advanced clustering approaches [157], [158], which can be divided into *early integration* and *late integration* approaches. While early integration approaches first concatenate the data of different omics layers and then cluster the merged data, late integration methods find patterns in the features of each layer separately, which can be combined as input for a regression or classification [84]. For early integration, data can either be concatenated by omics-features (rows) or conditions (columns). `OmicsTIDE` applies an early integration approach by concatenating two omics data sets by condition and clustering the concatenated matrix.

Commonly, the results of the integration methods are visualized in node-link diagrams or in trend visualizations, such as heatmaps and profile plots. The tool `3Omics` clusters up to three different omics layers, e.g. transcriptomics, proteomics, and metabolomics data hierarchically and visualizes the results as a clustered heatmap [21]. Alternatively, it creates correlation networks as node-link diagrams. A similar heatmap visualization has been implemented in the tool `multiSLIDE`, which combines two heatmaps side-by-side comparing transcriptomics and proteomics data [159]. While heatmaps represent one of the most commonly used approaches for visualizing abundance data, they can become huge when analyzing a large

number of genes. Due to their size and because of the usage of the color encoding, trends may become difficult to determine [160].

`Paintomics` follows an alternative integration approach for multiple omics layers by associating the omics-features, such as genes, proteins, and metabolites with their respective `KEGG` pathways and conducting pathway enrichment [152]. Each pathway can be analyzed in detail where the major trends of the associated features in different omics layers are displayed. However, it does not show to what degree the single omics-features contribute to the final trend and trends cannot directly be compared between pathways.

For trend comparison different strategies have been developed. For instance, an approach to visualize and compare trends was demonstrated in a study on the comparison of the transcriptomes of *Arabidopsis thaliana* and *Zea mays* [161], where trends in orthologous genes in leaf development were determined and compared. For the visualization of trends, the authors use profile plots, which are compared between the organisms using a table showing the orthologous genes overlapping between the trends. This approach provides a good overview of the trends in the two data sets. However, for the hierarchical clustering for categorization into discrete trends, clusters have to be separated manually. Moreover, as clustering was done independently for each data set, there is no inherent concept for trends having the same or different trajectories in the data sets. Thus, genes can not be classified as following the same or different behaviors.

Despite the fact that many tools have been developed to integrate multi-omics data, approaches integrating the data computationally, while keeping the integration process transparent using an exploratory visualization are rare. Overcoming this limitation was the main motivation for the development of `OmicsTIDE`.

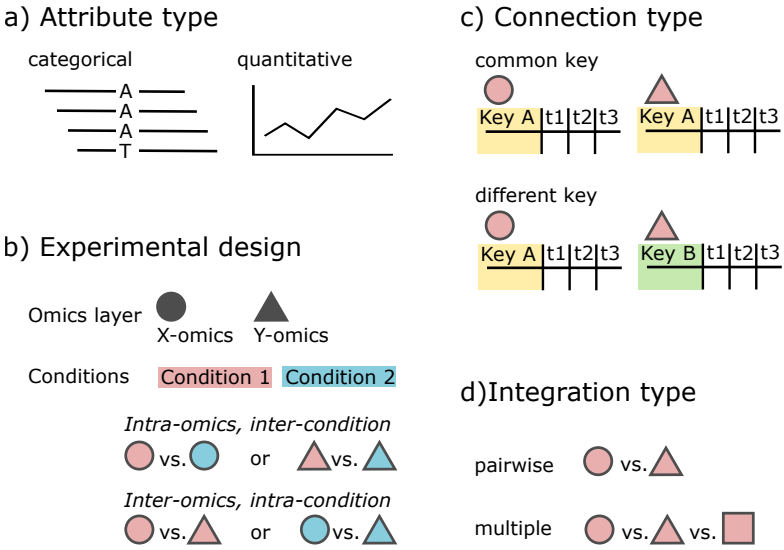


Figure 5.1: Omics data can be classified in different ways. (a) The *attributes* of omics features are either of categorical (e.g. mutations) or quantitative (e.g. transcript/protein levels) type. (b) Omics data analysis can also be classified by the *experimental design*. The analysis typically includes either the comparison of different conditions in the same omics layers (*inter-condition* and *intra-omics*) or the comparison of data from different omics layers within the same condition (*intra-condition* and *inter-omics*). (c) The combined analysis of omics data sets can be classified by whether the data sets can be joined by a common *key attribute* or not. (d) The *integration* of different omics layers can either be done in a pairwise fashion or by directly comparing multiple layers at once.

5.4 Classification of Omics Data

To identify the requirements for a novel multi-omics visualization tool and to create an abstract representation of the data, we developed a classification scheme. This classification builds the requirement framework for `OmicsTIDE`. First, omics data can be classified by the *attribute type*, which can either be categorical, such as the different bases of SNPs in genomics research, or quantitative, such as expression levels of genes, proteins, or metabolites (Figure 5.1a).

Secondly, comparative omics experiments can be classified by experimental design, which depends on the research goals (Figure 5.1b). Experiments are often performed within an omics layer (*intra-omics*) and between different conditions (*inter-condition*). Alternatively, omics experiments can include multiple omics layers (*inter-omics*) studying the same biological condition (*intra-condition*). In the case studies section we show a use case with an *inter-omics* and *intra-omics* experimental design. The design is applied to find differences in the transcriptomes of two strains under the effect of phosphate depletion (*intra-omics, inter-condition*), and to study how these differences are reflected in the proteome (*inter-omics, intra-condition*) [141].

When the *inter-omics* approach is chosen as experimental design, the connection between data sets can be created based on common keys with which the data sets can be combined or compared (Figure 5.1c). If the data sets do not share keys, a direct comparison cannot be conducted.

For *inter-omics* experimental designs, the decision on the number of omics layers determines the subsequent downstream analysis steps (Figure 5.1d). In order to study a given biological question it might be sufficient to compare two omics layers. More complex questions might require more than two omics studies (*multi-omics*) for a more powerful analysis to find specific patterns in the integrated data sets.

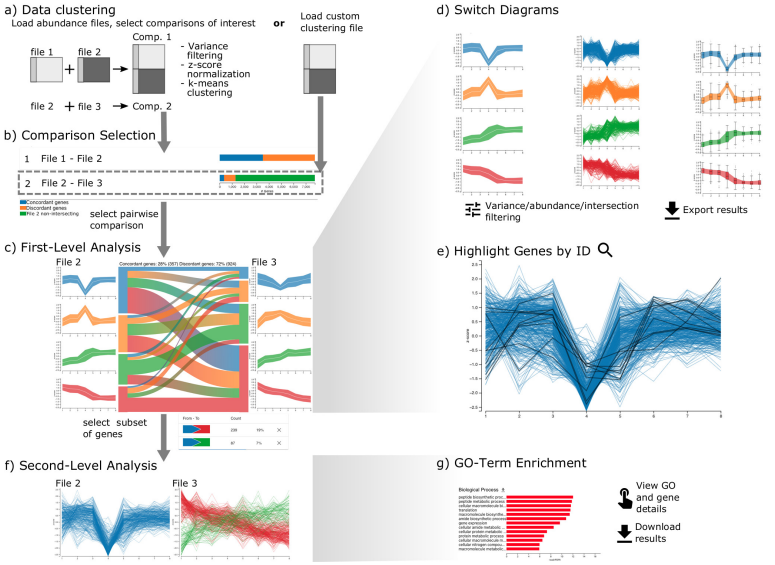


Figure 5.2: Basic workflow using `OmicsTIDE`. (a) Either multiple abundance files or a single custom trend-comparison file can be uploaded. (b) An overview of all conducted pairwise trend comparisons is shown as horizontal stacked bar chart showing the count of genes either being found in both compared files (intersecting) or only in one of the two files (non-intersecting). The number of intersecting genes in the bar chart is further categorized by either following a concordant (blue bar) or discordant trend (orange bar) in the two compared files. (c) After selecting a pairwise comparison in the overview visualization the data can be analyzed in the first-level analysis, consisting of a Sankey diagram comparing trends in both abundance files. (d) Users can switch between different trend diagrams, (e) highlight genes using gene IDs and filter data by abundance and variance. (f) For a more detailed analysis subsets of genes can be analyzed in the second-level analysis in detailed profile plots showing the expression of single genes (y-axis) across conditions (x-axis). Moreover, second-level analysis includes (g) viewing the NCBI entries of single genes and GO term enrichment analysis in a bar chart. The x-axis corresponds to the $-log_{10}(FDR)$ (*False Discovery Rate*) values and the y-axis corresponds to the significantly enriched GO-tems ($FDR < 0.05$). The color of the bars encodes for the term being overrepresented (red) or underrepresented (blue).

5.5 Method

Based on the classification scheme and our survey of related work, we developed four goals for the development of `OmicsTIDE`:

1. **Interpretability**: Provide a balance between sophisticated integration and interpretability for users.
2. **Applicability**: Focus on data that is widely available. Proteomics and Transcriptomics data are some of the most commonly produced data types.
3. **Overview-Detail**: Provide an overview of the integration, while also allowing detailed analysis of small subsets of the data or even single omics-features.
4. **Functional analysis**: Summarize subsets of omics features functionally to gain insights about the underlying biological context.

`OmicsTIDE` employs pairwise integration of two omics layers (*inter-omics*, *intra-condition*) and the comparison of two data sets within an omics layer (*intra-omics*, *inter-condition*) using a simple concept of clustering the data into trends of omics features following the same trajectory (Goal 1). `OmicsTIDE` only requires data from one or two omics layers to produce meaningful results (Goal 2). The tools offers a detailed analysis of omics-features of interest identified in the overview visualization (Goal 3). Groups of omics-features can be analyzed functionally using GO-term enrichment (Goal 4).

`OmicsTIDE` computes and visualizes trends for two-dimensional experimental designs. The first dimension is represented by the data sets that are compared, which can be from one or two different omics layers. The second dimension is represented by conditions that need to be consistent across data sets, such as time points or environmental conditions.

The central idea of the visualization approach is to compare trends occurring in two omics data sets using a Sankey diagram, which is a graphical representation of flows between sets. The trends of the different data sets are visualized adjacent to the *nodes* of the Sankey diagram. The height of the nodes encodes for the number of

genes found in the trends, while the thickness of the bands (*links*) between the nodes encodes for the number of genes that either show the same trends (concordant trends) or different trends (discordant trends) in the two data sets.

Data sets are compared in three major steps referred to as *comparison selection*, *first-level analysis* and *second-level analysis* (Figure 5.2). The separation of the analysis is reflected in the dynamic tab-based design of `OmicsTIDE`, with which new tabs corresponding to the respective analysis steps can be added. With this design, choices made in any tab can be reviewed, refined, or removed.

During all steps of the analysis intermediate results can be exported in CSV format for downstream analysis with other tools. For easy sharing of the visualizations, they can be exported in PNG or PDF format.

5.5.1 Data Loading and Comparison Selection

`OmicsTIDE` offers two distinct data input options in form of abundance files or a custom clustering file (Figure 5.2a). Users can load multiple abundance files that are compared in a pairwise fashion and clustered by `OmicsTIDE` to obtain trends. Each abundance file containes genes (rows), conditions (columns) and normalized abundance (cells). After choosing abundance files, users can choose to restrict the analysis to variant genes by removing genes based on the percentile range of their variances across different conditions. This reduces the formation of trends that are influenced by low-variance genes. Moreover, if more than two data sets are to be analyzed, the pairwise comparisons of interest can be selected. For fast exploration of the data `OmicsTIDE` uses k-means. For this, the number of trends to be derived from the data has to be chosen in a range between 2 and 10. To make all data sets comparable, `OmicsTIDE` applies z-score normalization to each data set prior to clustering. To guarantee flexibility in the choice of clustering algorithms, users can upload their own clustering results for a pairwise trend comparison.

For each pairwise combination, `OmicsTIDE` conducts two separate trend comparisons: One for the genes found in both data sets (intersecting genes) and one for the genes found only in one of the two

data sets (non-intersecting genes). While non-intersecting genes are clustered separately for each data set, for the intersecting genes, `OmicsTIDE` makes use of an *early integration* approach by first combining the two data sets using the shared conditions and then clustering the combined matrix with `k-means++` [162]. Therefore, each gene is associated with two clusters, one for each data set. With this approach, the genes can easily be classified as following concordant or discordant trends, which represents an intuitive concept for users.

An overview visualization helps users choose a comparison for the trend exploration in the first-level analysis (Figure 5.2b). The comparisons are visualized using stacked bar charts showing concordant, discordant, intersecting, and non-intersecting genes and are thus providing a useful rationale to select a comparison of interest.

5.5.2 First-level Analysis: Trend Exploration

The first-level analysis tab provides a detailed visualization of the selected trend comparison of intersecting genes in a Sankey diagram together with profile plots (Figure 5.2c, Figure 5.3), as well as a sidebar containing controls for interactive features (Figure 5.3). The Sankey diagrams shows the size of the gene sets corresponding to the trends (nodes) and how many genes are contained in the trend intersections, i.e. shared between the trends of the two data sets (links). The trends, nodes, and links are colored using a set of categorical colors. A color-gradient is used for the bands transitioning between the colors of the connected nodes. The gradient is inverted and shows the color of the left node on the right side and vice versa to simplify the identification of trends in one data set connected to a single trend in the other data set by looking at the corresponding node. A summary of the comparison is displayed at the top of the visualization, showing the number and percentage of concordant and discordant genes.

By default, the trends are visualized using *centroid profile plots*, which provide an overview by showing the centroid line as well as the standard deviation of the trend as a band. Alternatively, using the controls sidebar users can choose profile plots, where each

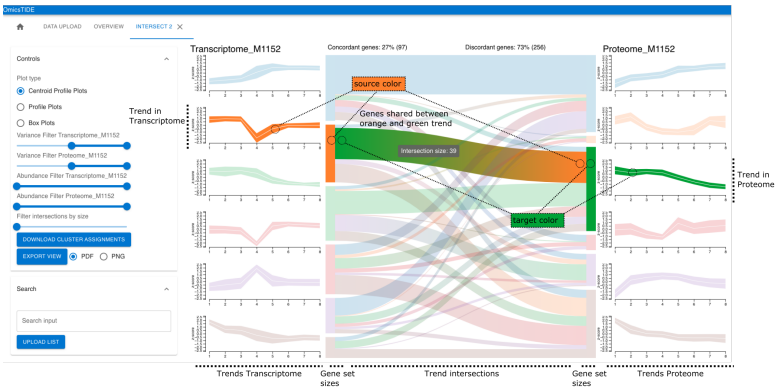


Figure 5.3: Screenshot of `OmicsTIDE`. `OmicsTIDE` employs a tab-based design for easy exploration of multiple comparisons. The selected tab (Intersect 2) corresponds to the intersecting analysis of the second comparison. The visualization consists of centroid profile plots representing trends of genes as well as a Sankey diagram showing the intersections of trends between the data sets. Each trend represents a set of genes belonging to a cluster and a data set. The trends are visualized as centroid profile plots and colored with a set of categorical colors. These colors are repeated in the nodes of the Sankey diagram showing the sizes of the gene sets corresponding to the trends. The bands (links) of the Sankey diagram encode for the intersections between the trends. The band connecting the orange and green trend has been highlighted. The color gradient of the bands is inverted to allow a quick identification of all target nodes belonging to a source node. In the sidebar (on the left), interactive controls allow the user to change plot types of the trends, filter the data, export results, and search for genes.

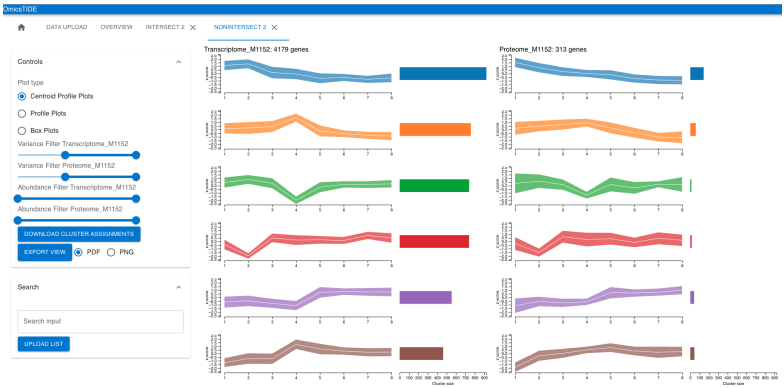


Figure 5.4: First level analysis of non-intersecting genes. For non-intersecting genes, trends can be visualized as centroid profile plots (as shown here), profile plots, and box plots. Gene set sizes are shown as horizontal bar charts, next to the respective profile plots.

gene is plotted as a line for a more detailed view on the composition of each trend (Figure 5.2d). This visualization is more suitable for a low number of genes since the visualization of a large number of gene profiles may result in overplotting. As a third option, the user can study the abundance variation per condition within a trend in more detail using box plots. In addition to the analysis of intersecting genes, the trend visualizations are used for analyzing non-intersecting genes. Since non-intersecting genes do not share identifiers, they are not connected with a Sankey diagram but only show trend visualizations and horizontal bar charts to show sizes of the gene sets corresponding to the trends (Figure 5.4).

The nodes and links in the Sankey diagram can be hovered to study the single trends between the two data sets in more detail. When a node is hovered, all connected links are highlighted by reducing the opacity of other elements (*focus-on-hover* strategy). Hovering over elements in the visualization updates the detail diagrams accordingly. This update is facilitated via an animated transition to visually link the hovering and the data update. Moreover, all concordant or discordant intersections can be highlighted by hovering over the concordance/discordance summary.

Users can check their own hypotheses about gene sets of interest, such as genes from specific pathways, and analyze their behavior across trends and data sets by highlighting genes of interest by their gene IDs (Figure 5.2e). Users can directly type in one or more gene IDs into a text field or upload a text file with gene IDs. The profiles of the genes in the diagram corresponding to the given IDs in the query are marked in black.

To study the effects of the variance or abundance levels of genes on the trends, `OmicsTIDE` can dynamically filter data by the percentile ranges of the variance or the median abundance of the genes during first-level analysis. The variance and the median abundance and their respective percentiles are calculated prior to z-score normalization. The variance filtering in the first-level analysis can be applied as an alternative or in addition of the variance filtering provided when loading the data. In contrast to the variance filtering before loading the data, which is considered a preprocessing step, the filtering in the first-level analysis allows users to explore different ranges of variances quickly. Additionally, users can filter intersections by size to remove small intersections from the visualization and to thus reduce visual clutter.

5.5.3 Second-level Analysis: Detailed Trend Analysis

Sets of genes corresponding to trends or the intersection of trends can be analyzed in detail to find, for example, enriched functions, thus implementing our fourth goal. `OmicsTIDE` allows users to select either links or nodes in the visualization to extract subsets of genes. A table placed in the controls side bar shows the source node and the target node of each selected link as well as the number and percentage of the corresponding genes (Figure 5.2c, bottom). Thereby, users can compare the actual numbers of genes corresponding to a link.

Selected genes can then be analyzed in detail in the second-level analysis (Figure 5.2f). Users can study gene sets on the single-gene level by hovering the single gene profiles and accessing information of an individual gene by clicking and being redirected to the corresponding `NCBI` entry. Furthermore, the gene subsets can be

analyzed in a Gene Ontology (GO) context (Figure 5.2g). Thereby, users can find GO terms that are enriched in in the selected subset and form hypotheses about the regulatory processes causing the patterns. The `PantherDB` API is used to perform GO enrichment for the three main GO categories *molecular function*, *biological process*, and *cellular component* using *Fisher's exact test* and a multiple test correction with *False Discovery Rate (FDR)* [163]. Users can either choose to use the whole genome as a background for the enrichment, or only the genes contained in the current first-level analysis, as certain classes of proteins can, for example, be underrepresented in proteomics data. The results are visualized as horizontal bar charts where length encodes for the negative logarithm of the *FDR* and color encodes for the term being overrepresented or underrepresented to allow users to quickly identify the most significant results. Hovering the single bars will show a tool tip with more detailed information on the corresponding GO term. Clicking on the bar will open a tab with further information about the GO term on `Amigo` (http://amigo.geneontology.org/).

5.5.4 Implementation

`OmicsTIDE` is a web-based client-server application that uses *Python* for complex computations, such as the trend determination via clustering in the back-end and `Flask` for communication with the front-end [164]. The libraries React (https://reactjs.org/) and Mobx (https://mobx.js.org/) are used for the application structure of the front-end and state-management. The `JavaScript` library `D3.js` is used for creating the visualizations and animation [146]. The application styles are created using Material-ui (https://mui.com/). The source code of `OmicsTIDE` is available at https://github.com/Integrative-Transcriptomics/OmicsTIDE2.0.

5.6 Case Studies

To demonstrate the applicability of the pairwise trend comparison approach in `OmicsTIDE`, we conducted two case studies. In the first case study, we show how the combined analysis of transcriptomics

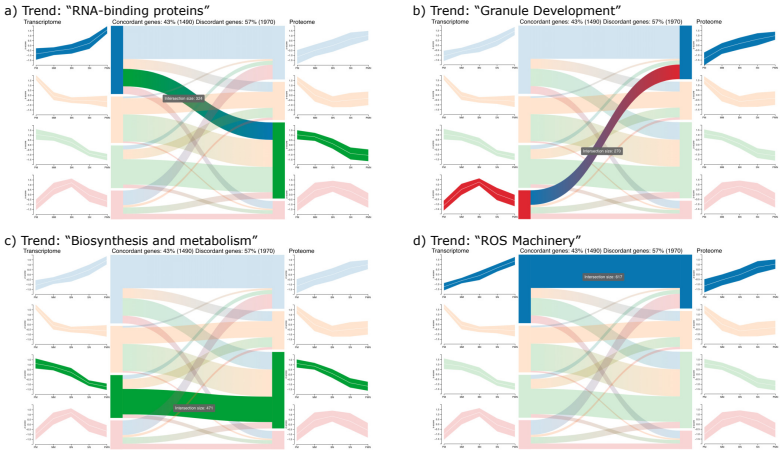


Figure 5.5: First-level analysis: Interactively studying the trend comparison between the transcriptome and proteome during blood cell differentiation [165] by hovering reveals several concordant and discordant trends. (a-d) Four trends found by `OmicsTIDE` were associated with functionally annotated modules found in the study. The tooltip displays the number of contained genes.

and proteomics data can be used to extract biologically relevant concordant as well as discordant trends with few clicks only. The second case study combines two pairwise trend comparisons to extract information from both, different experimental conditions and different omics layers to demonstrate the synergy that can be achieved by `OmicsTIDE`.

5.6.1 Blood Cell Differentiation in Bone Marrow

Neutrophils are an essential part of the human immune system. They are differentiated in the bone marrow and released to the bloodstream. The regulation of the neutrophil differentiation is subject of the first case study, examining granulopoiesis *in vivo* [165]. The experimental design uses both transcriptome and proteome data from the five differentiation stages (pro)myelocytes (PMs), metamyelocytes (MMs), immature neutrophils with band-shaped

nucleus (BN), mature neutrophils with segmented nucleus (SNs) and the actual peripheral mature neutrophils (PMNs). Here, we show how `OmicsTIDE` can be used to efficiently reproduce the findings made by Hoogendijk, Pourfarzad, Aarts, *et al.* [165] by exploring the trends between the two omics layers. The data was taken from the supplementary material that contained quantified transcripts and proteins in the form of *Fragments Per Kilobase Million* (FPKM) and imputed log_2 *Label Free Quantification* (LFQ) measures, respectively. The data included four replicates for transcripts and three replicates for proteins for each of the five conditions. The analysis was performed on the mean values of all biological replicates for each condition.

To explore the trends shown by the transcriptome and the proteome of different blood cell types, the selection of $k = 4$ initial clusters resulted in clearly distinguishable trends that are shown as centroid profile diagrams for either data set resulting in 16 trend intersections (the maximum possible for $k = 4$) (Figure 5.5). Similarly, Hoogendijk, Pourfarzad, Aarts, *et al.* [165] extracted 12 *modules* from the data, which represent combinations of trends in the transcriptome and proteome. The authors combined the modules based on their main trajectories, such as concordant increasing, concordant decreasing, and increasing in the transcriptome while decreasing in the proteome. They classified the combined modules based on GO enrichment and the enrichment of specific database entries.

We visually identified four combined modules using `OmicsTIDE` by hovering the single links in the Sankey diagram (Figure 5.5). As a next step we applied GO-enrichment analysis using `OmicsTIDE` to confirm the classifications. Fitting GO-terms were found for "RNA-binding protein" (GO:0003723, RNA-binding Figure 5.5a) and "Granule Development" (GO:0042581, specific granule, Figure 5.5b). For "biosynthesis and metabolism" (Figure 5.5c) we found a number of terms that are involved in these processes (for example GO:1901566, GO:0009205, GO:0006754) rather than finding a single fitting term. A reason for this could be that "biosynthesis and metabolism" is a very broad category involving a lot of GO-terms.

Table 5.1: Comparison of modules created by Hoogendijk, Pourfarzad, Aarts, *et al.* [165] with trend intersections created in OmicsTIDE.

Trend transcriptome Trend Proteome Function	Increasing Increasing ROS Machinery	Decreasing Decreasing Biosynthesis and Metabolism	Increasing Decreasing RNA-binding proteins	Curved Increasing Granule development
# genes Hoogendijk et al.	621	1320	212	192
# genes *OmicsTIDE*	617	1439	508	270
Intersection	486	1131	177	132

The annotation "ROS machinery" could not be solely reproduced using the GO-enrichment of `OmicsTIDE` (Figure 5.5d), since the authors used a combination of GO enrichment and manual enrichment using other databases specialized on the annotation of human proteins. To confirm that the underlying gene sets are similar we compared them manually. Since the authors grouped the modules into broader categories the sets of concordant genes stemming from both decreasing trends were merged for `OmicsTIDE` as well. Overall, `OmicsTIDE` produced 617 increasing concordant genes, while 621 were found in the blood cell study with an overlap of 486 genes (Table 5.1). This indicates that `OmicsTIDE` includes a trend intersection containing a gene set similar to the one annotated with "ROS machinery". Similarly, of the 1320 decreasing concordant genes, 1131 could be found in similar patterns in `OmicsTIDE` (yellow and green trend, total of 1439 genes). The other modules compared were much smaller and we found more genes in `OmicsTIDE`. Yet, we could find more than 70 % of the genes of each module.

5.6.2 Transcriptome and Proteome Time Series Data Set of *Streptomyces coelicolor*

To demonstrate how *inter-omics* as well as *intra-omics* analysis can be combined using `OmicsTIDE`, we re-analyzed the data sets of a study exploring two *Streptomyces coelicolor* strains with respect to changes in their metabolisms under phosphate-starving growth

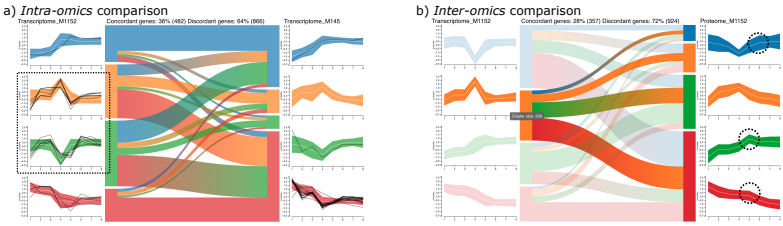


Figure 5.6: Combination of *intra-omics* and *inter-omics* analysis for *Streptomyces coelicolor*. (a) First-level analysis of the transcriptomes of the strains M1152 and M145 after focusing on genes with a median abundance above the 80th percentile. A custom list containing genes involved in the metabolic switch to nitrate respiration under phosphate starvation has been highlighted (black lines). The green and orange trend show exactly inverse patterns. (b) First-level analysis of M1152 transcriptome (left) vs. M1152 proteome (right) trend in the transcriptome. Only a small fraction of genes follows the same trend in the proteome. Three trends in the proteome have a peak at a later timepoint than the peak of the orange trend in the transcriptome.

conditions in a time-course experiment [141]. The *Streptomyces coelicolor* strains M145 and M1152 were used to study the role of *biosynthetic gene clusters* (BGCs) for the production of antibiotics. M1152 is a genetically-engineered derivate of the M145 wild-type strain that was subject to the deletion of different BGCs [166]. For both strains samples were collected at eight timepoints. Phosphate was depleted between timepoint 3 and timepoint 4. For each of the time points, three biological replicates were generated for each omics layer. Both, transcriptome and proteome data was first quantified and log_2-transformed. Next, the data was normalized by an *intra-strain* and *intra-omics* quantile-normalization across all replicates. Finally, the mean of the three replicates was calculated.

In `OmicsTIDE` the four data sets (M145 transcriptome, M1152 transcriptome, M145 proteome, M1152 proteome) were loaded resulting in six pairwise trend comparisons. For the k-Means clustering $k = 4$ was chosen since it produced the most clearly distinguishable trends. We first focused on the comparison of two different strains across a single omics layer (M1152 transcriptome vs. M145 transcriptome)

to find differences on the transcript level. The insights from this first pairwise comparison were then used to study whether these insights are reflected in the proteome of the mutant strain. In particular this inter-omics comparison had not been subject to the study of Sulheim, Kumelj, Dissel, *et al.* [141].

Intra-Omics: M1152 transcriptome vs. M145 transcriptome

The *intra-omics* comparison of the M1152 transcriptome and the M145 transcriptome revealed a total of 7,904 genes that appear in both data sets, whereof around 55 % follow concordant trends (data not shown). After applying the abundance filtering to focus on genes with a high median abundance of above the 80th percentile in both data sets the shape of the trends becomes clearly visible (Figure 5.6a). Interestingly, the centroid profile plots show that the green trend and the orange one show the exact inverse trend in the M1152 transcriptome. The same can be observed for the blue trend and the red trend. The inverse behavior of the trends is also partly reflected in the M145 transcriptome. However, about 64 % of the genes show discordant expression trends, indicating that the effect of phosphate depletion on gene expression differs between the two strains.

These findings were investigated in more detail by combining the trend comparison with information on genes involved in the metabolic switch to nitrate respiration under phosphate depletion [167]. The corresponding gene IDs were highlighted in the profile diagrams (Figure 5.6a). Intriguingly, all but one highlighted gene follow the same trend in the M145 transcriptome and are downregulated until timepoint four (red trend), followed by an up-regulation. In the M1152 transcriptome the genes are distributed across three trends, most strikingly the green and orange trend where the highlighted genes show a peak at time point 4, but go back to the expression level observed before the depletion.

Inter-Omics: M1152 transcriptome vs. M1152 proteome

We investigated how the patterns of trends co-ouccuring with phosphate depletion in the M1152 transcriptome are reflected in the

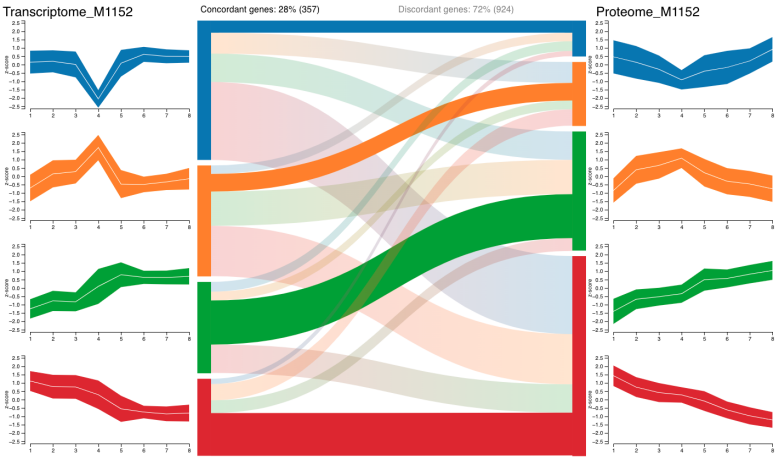


Figure 5.7: The inter-omics trend comparison between the transcriptome and proteome of S. coelicolor M1152. The concordant intersections are highlighted. While the trends with prominent peaks (blue and orange) of the transcriptome show little concordance with the Proteome, the more constantly increasing (green) or decreasing (red) trends show higher inter-omics concordance.

corresponding proteome (Figure 5.6b, Figure 5.7). The peaks in the blue and orange trends of the transcriptome appear shortly after phosphate depletion and the trends show low concordance with trends in the proteome. In contrast, the green and red trend show constantly increasing and decreasing behaviors with high concordance in the proteome. Therefore, we can conclude that the transcriptomic trends with peaks associated with phosphate depletion are not directly evident in the proteome, while the constant trends are more concordant in their behavior across the two omics layers. However, when hovering over the orange trend, we detected that the remaining three trends in the proteome all share a small peak at a later time point (Figure 5.6b). Since this small peak appears at a later time point than the peak in the orange trend of the M1152 transcriptome, it could be further investigated whether this suggests a time-delayed translation of the protein cognates.

5.7 Discussion

In this paper, we present `OmicsTIDE`, an easy to use analysis and visualization tool for the concurrent exploration of multi-omics data implementing our goal of interpretability (Goal 1).

In the context of developing `OmicsTIDE` we also devised a classification system for multi-omics data, which offers an underlying framework for our tool, but may also serve useful for future developments in this field. The interactive trend comparison in `OmicsTIDE` using the concept of concordance and discordance emphasizes the similarities and differences between two omics data sets. With this, it marks an innovation compared to other tools that mainly aim to integrate a large number of omics data sets to derive a combined pattern. It should be noted that the pairwise analysis and the multi-omics integration are not mutually exclusive ways of analysis, but rather complement each other.

`OmicsTIDE` uses a Sankey diagram to compare trends across data sets. With this visualization, concordance and discordance between trends can be intuitively explored. The trends are either visualized using *centroid profile plots*, profile plots, or boxplots. While centroid profile plots visualize an overview of the profile, detailed

profile plots show every gene separately. With this detailed visualization it is easier to track the behaviour of single genes. Profile plots are especially useful if the order of conditions is inherent, such as time series [160]. In contrast, boxplots do not assume that the conditions are ordered and are, therefore, better suited for categorical data. Moreover, they focus on visualizing the distribution of values at each condition. This is especially useful to identify outliers or for assessing consistency across replicates.

To compute trends from multi-omics data, `OmicsTIDE` uses an *early integration* approach by first concatenating and then clustering the data. Currently, for the clustering `k-Means++` is applied in `OmicsTIDE`. In addition, `OmicsTIDE` can use any early integration clustering uploaded manually by the user. While we were able to show that applying k-means extracts the main trends which can clearly be distinguished, we plan to implement more sophisticated clustering algorithms, such as `dbscan` [168] or `iCluster` [169] in a future version. Such approaches might prevent biased trends, especially if the number of genes in one of the compared data sets is very high compared to the other data set. To counteract this bias, we analyze intersecting and non-intersecting genes separately, which guarantees an equal number of genes for both data sets in the intersecting analysis.

The ability of `OmicsTIDE` to extract and compare trends was demonstrated in two case studies using different experimental designs. In the first case study, the integrated analysis of transcriptome and proteome data shows that `OmicsTIDE` can derive the most important information in few steps leading to findings similar to the ones in the original study. These findings were further consolidated by a manual comparison of the genes extracted from the intersections of the trends in `OmicsTIDE` and the modules defined by the authors of the publication. Although the modules could not be reproduced perfectly in `OmicsTIDE` due to the much simpler clustering approach, between 70 % and 85 % of the genes found in the respective modules agreed with the trends identified in `OmicsTIDE`.

The second case study applies a more complex experimental design enabling an *intra-omics* as well as *inter-omics* comparison. As `OmicsTIDE` provides the option of combining different pairwise omics

data comparisons within a single analysis according to our third goal, trends could be analyzed in the *intra-omics* as well as the *inter-omics* comparison while keeping an overview of all involved data sets. The exploration of the Sankey diagram using the *focus-on-hover* strategy could show that the trends initially found in the intra-omics analysis (the transcriptome comparison) are also revealed in the proteome. In summary, the parallel analysis of *intra-* and *inter-omics* data in `OmicsTIDE` leads to easily interpretable expression trends and possible hypotheses.

In `OmicsTIDE`, we compare data sets using shared keys (e.g., gene IDs), which facilitates the comparative visualization of trends. In a future version, a pairwise comparison between omics layers not sharing keys and an advanced comparison of non-intersecting genes could be achieved by linking keys using meta-information, such as common pathway IDs.

With `OmicsTIDE` we present a tool for initial exploration and hypothesis generation, which complements advanced statistical or machine-learning methods. The choice of additional analysis methods depends on the generated hypothesis. Yet, in future versions, we plan to integrate methods for statistical validation of the extracted trends.

`OmicsTIDE` is designed in particular for biologists; its user interface creates clear default views that show the concordant and discordant patterns in omics abundance data in a pairwise manner. The simple input format (numerical matrices) leads to great flexibility in `OmicsTIDE` as it can perform *inter-omics* as well as *intra-omics* comparisons, thus allowing for example also the comparison of two transcriptomic, proteomic, or metabolomic data sets as well as the analysis of complex mixed-omics experimental designs.

Chapter 6

Visualizing Cohorts: OncoThreads

Multi-omics data is of more and more relevance in medical research. Especially in cancer research, many different types of data are studied for large cohorts to characterize the different types and stages of the disease. Defining subgroups of cohorts based on multi-omics data can help understand the underlying mechanisms and develop more targeted treatments.

This chapter includes previously published work on `OncoThreads` presented at ISMB 2021 [28]. For readability, figures originally included in the supplementary material have been included in the main text. References to the supplementary video, available in the supplementary material of the publication have been removed. The published correction about a figure doubled in the manuscript has been applied directly to the text.

OncoThreads: visualization of large-scale longitudinal cancer molecular data

6.1 Abstract

Motivation: Molecular profiling of patient tumors and liquid biopsies over time with next-generation sequencing technologies and new immuno-profile assays are becoming part of standard research and clinical practice. With the wealth of new longitudinal data, there is a critical need for visualizations for cancer researchers to explore and interpret temporal patterns not just in a single patient but across cohorts.
Results: To address this need we developed `OncoThreads`, a tool for the visualization of longitudinal clinical and cancer genomics and other molecular data in patient cohorts. The tool visualizes patient

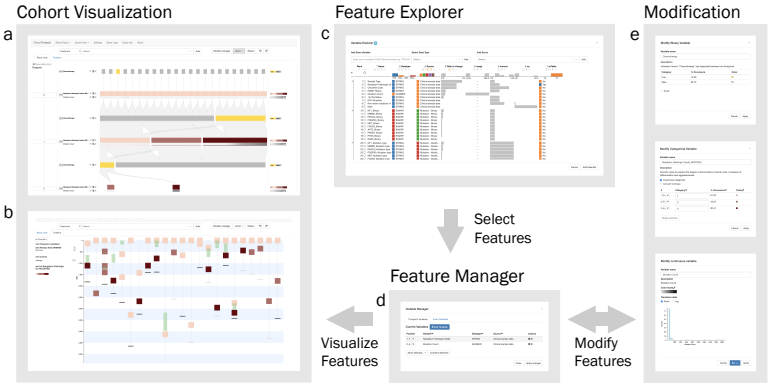


Figure 6.1: A schematic view of the components of `OncoThreads`. Molecular data can be visualized in two separate views, the block view (a), which aligns shared events of patients as blocks and the timeline view (b), which shows a timeline for each patient. Features of interest can be found and selected in the Feature Explorer (c) and added to the Feature Manager (d), which supplies them to the visualization. The application also enables feature modification using different types of transformations depending on the type of the feature (e).

cohorts as temporal heatmaps and Sankey diagrams that support the interactive exploration and ranking of a wide range of clinical and molecular features. This allows analysts to discover temporal patterns in longitudinal data, such as the impact of mutations on response to a treatment, e.g. emergence of resistant clones. We demonstrate the functionality of `OncoThreads` using a cohort of 23 glioma patients sampled at 2-4 timepoints.
Availability: Freely available at `http://oncothreads.gehlenborglab.org`. Implemented in Javascript using the cBioPortal web API as a backend.

6.2 Introduction

New profiling technologies, including next generation sequencing, have significantly expanded our molecular understanding of cancer.

Projects such as The Cancer Genome Atlas (TCGA), the International Cancer Genome Consortium (ICGC) and the Human Tumor Atlas Network (HTAN) have set out to comprehensively characterize tumor samples by generating multi-omic data sets which support the identification of molecular subtypes and new, targeted treatment opportunities [36]–[38].

These projects have sparked the development of new tools to visualize and explore these large data sets, including: the `cBioPortal` for Cancer Genomics, a widely used platform for the analysis and visual exploration of cancer genomic data sets [104], [105]; genomic browsers like `UCSC Xena` [103] and others [24]; and cohort visualization tools like `StratomeX` [154], [170], [171].

Despite the advancement of cancer-specific visualizations and portals, temporal visualizations are often lacking. `cBioPortal` offers a temporal view for individual patients which supports a range of data types, including procedure and treatments [104], [105]. Another temporal visualization is the "fishplot", which shows the development of tumor subclones in an individual over time [172], [173]. However, neither approach scales well for entire cohorts, as subclone evolution is highly individual and cohort visualizations with individual patient timelines become cluttered even for a small number of patients and time points. Tools like `EventFlow` [174] and `DecisionFlow` [175] tackle this problem by aligning shared events in cohorts in blocks with transitions between events displayed as flows. Another approach has been implemented by Perer et al., where events in a cohort are grouped into timepoints and displayed in matrices showing the co-occurrence of events [176]. While these approaches are useful for analyzing event sequences, as well as for selecting and comparing cohorts [177], they do not integrate multiple features for events, such as mutation data and expression data for sample collection events. A more flexible block-based technique is `Domino`, which is a visualization technique for the creation of multiple connected visualizations [178]. Despite not being developed specifically for temporal data, a wide range of temporal visualizations can be implemented and modified directly in the tool. However, due to its high flexibility and the novel underlying concept, it is difficult to apply for users who are not visualization experts.

OncoThreads was designed for cancer researchers and developed to address the lack of temporal cohort visualization tools, which specifically integrate multiple molecular data types and clinical data. OncoThreads provides exploratory visualizations of longitudinal cancer molecular data across patient cohorts and supports a wide range of biological data types, including mutations, copy number alterations, mRNA expression and protein expression. Furthermore, OncoThreads offers a temporal cohort visualization based on heatmaps and Sankey diagrams as well as a timeline overview for all patients. Moreover, it provides a feature explorer to discover features of interest - variables that are defined for each patient and timepoint, such as tumor stage or mutation burden - and feature modification in order to adjust their visual representation and facilitate interpretation. We demonstrate the ability of OncoThreads to enable the exploration of longitudinal cancer molecular data in a comprehensive case study with a cohort of 23 glioma patients (Section 6.4). Moreover, we assess the usefulness of the design sprint approach [179] for the development of exploratory visualizations.

6.3 Meterial & Methods

6.3.1 OncoThreads Overview

OncoThreads enables researchers to dynamically visualize longitudinal clinical and molecular data across an entire patient cohort, allowing for the identification of patterns in cancer evolution. For example, researchers can visualize tumor stage, mutations, mRNA expression levels or tumor mutation burden at multiple timepoints for an entire patient cohort. The application consists of several components for the selection of features and temporal visualization (Figure 6.1).

OncoThreads displays time as a vertical flow from top to bottom in order to accommodate large patient cohorts, which are presented horizontally. The selected features can be visualized in two separate views. In the block view, samples and events are aligned in blocks in order to show general event patterns of the cohort over

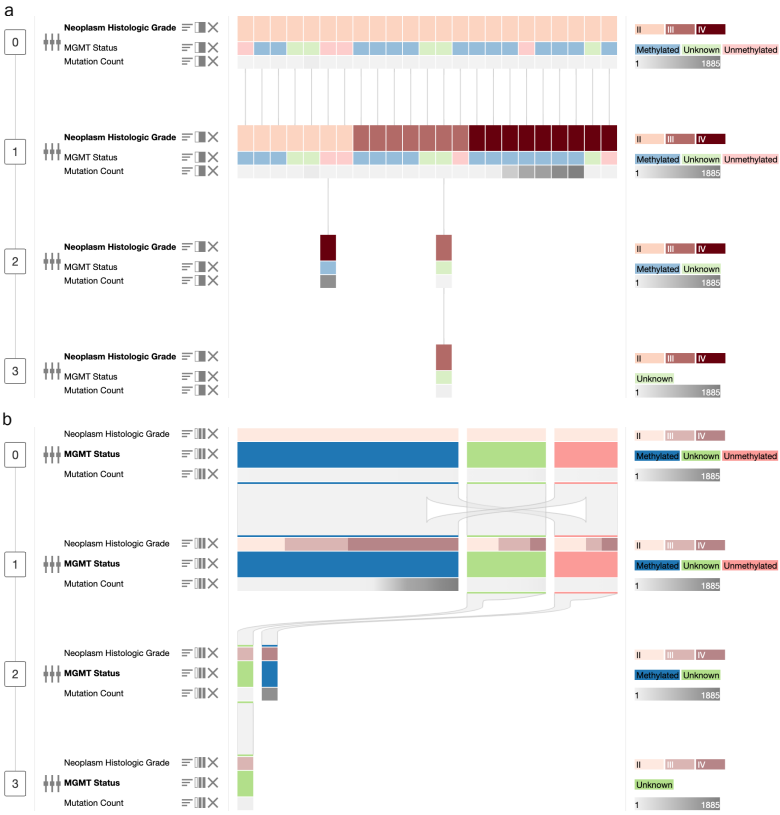


Figure 6.2: Visualization operations in `OncoThreads`. Blocks represent timepoints, which are ordered vertically. (a) Heatmap view with multiple sample-level clinical features (Mutation Count, MGMT Status, and Neoplasm Histologic Grade). Patients are connected by lines. Multidimensional sorting is applied to the second timepoint, which is primarily sorted by Neoplasm Histologic Grade, while the secondary order is given by the other features. Patients in the other blocks have been aligned based on the order of this timepoint. (b) The same data with all blocks grouped by MGMT Status. Grouped blocks show proportions of patients instead of single patients. Within the primary grouping (MGMT status) the other features are grouped as well. Bands show the proportion of patients transitioning between feature values of two blocks.

time (Figure 6.1a). The timeline view shows a timeline for each patient reflecting the actual temporal distance between samples and events (Figure 6.1b). A user can alternate between these two views as data is explored. In order to keep track of the exploration, every action is saved in an accessible log and undo/redo functionality is provided. Additionally, users can export the current view, including detailed metadata about the displayed features, in multiple file formats (PNG, PDF or SVG).

Data can either be loaded using the `cBioPortal` API or local files. With the Feature Explorer, features can be ranked and selected according to attributes, such as their variability over time (Figure 6.1c). Additionally, features can be transformed in the Feature Manager, for example, to change a feature's color scale or to convert a continuous feature to an ordinal feature by binning or to aggregate genes into gene sets (Figure 6.1d,e).

6.3.2 Block View

The main visual element of the `OncoThreads` cohort visualization is a block. `OncoThreads` supports two types of blocks: timepoint blocks and event blocks. A timepoint block represents the samples of a patient cohort at a certain timepoint with associated clinical, genomic, or other molecular data (for example, samples acquired at initial and recurrent surgeries, or prior to and following a therapy). An event block represents events that occur between two timepoints (for example, treatment with a drug). Timepoint blocks are always visible, while event blocks can be added as desired; when both are visible, event and timepoint blocks alternate (Figure 6.1a, also see Section 6.4). The rows of a block represent a set of features. Upon loading a study, data within the blocks is visualized as a heatmap. Data within blocks can be rearranged to explore the data by sorting the entire heatmap with respect to a feature at a specific timepoint, or transforming it into a Sankey diagram by grouping.

Sorting enables the exploration of the distribution of values of a feature. Each block can be sorted individually with respect to a feature (called the primary feature). Since sorting may change the order of the patients to be different across timepoints, the connecting

lines are curved and may cross. In order to eliminate crossing lines, the patients can be realigned with respect to the patient order in any of the blocks (Figure 6.2a). Moreover, we also implemented multidimensional sorting, which sorts based on multiple features at once. When a block is sorted repeatedly by different features, the previous order of patients is retained and applied in case of ambiguities. This can be seen in Figure 6.2a, where the second timepoint is sorted by all three features.

The block view visualization can be transformed iteratively into a Sankey diagram by grouping timepoints to analyze the data as groups of patients rather than individual patients (Figure 6.2b). A grouped block shows information about the proportions of patients based on the primary feature, rather than showing individual patient data (see also the case study in Section 6.4). It therefore represents an aggregated view, while the heatmap shows the data in more detail. Due to the independent grouping and ungrouping of blocks, detail can be viewed selectively for certain timepoints, while others stay grouped and show proportions. Furthermore, grouping is especially useful for large cohorts since it might not be possible to visualize the entire cohort as a heatmap depending on the screen width.

If the primary feature is categorical, the proportions in grouped blocks are displayed as horizontal bars with widths corresponding to the size of the proportion. The proportions and distributions of other features are shown within the groupings of the primary feature to allow a comprehensive comparison of the compositions of different grouped blocks. Values of continuous features are summarized by visualizing their distributions using color gradients or boxplots. For continuous features many patients have unique values, which would lead to one patient per group. Consequently, a continuous feature has to be binned before grouping to transform it into a categorical feature as described in Section 6.3.4. A Sankey diagram is created whenever two adjacent blocks are grouped. The connection between blocks changes to bands showing the fraction of patients transitioning between the proportions of the blocks. To highlight that the bands originate from the primary feature and not from the

last row of the grouped block, the colors of the primary feature are repeated as a proxy at both ends of the connections (Figure 6.2b).

By default, patient samples are aligned with the first available timepoint for each patient as the first timepoint in the visualization. However, a cohort may have variability in the first available timepoint, or it may be of interest to analyze a cohort relative to an event instead, such as the administration of a treatment. Therefore we implemented flexible timepoint alignment. Patient columns can be selected in an ungrouped block and moved up or down using a context menu. Section 6.4 shows how this functionality is applied in a sample data set.

In order to track a subset of patients in the visualization, `OncoThreads` allows a user to select individual patients as well as groups of patients. The selected patients or patient groups are highlighted in all blocks and bands allowing the user to gain an understanding of the composition of a subset of patients in all blocks simultaneously.

6.3.3 Timeline View

In the timeline view, data is visualized as a series of adjacent vertical timelines, one timeline for each patient. Users can switch between the block view and the timeline view to analyze different aspects of the data. The timeline view can address questions such as the relationship between the duration of a therapy and time to progression. In this view, only one feature is displayed for each sample. Different events are encoded using different colors, and the duration of an event is encoded by the length of the bar (Figure 6.1b). Similar to the block view, patients can be selected interactively. Selected patients are retained in both views. Therefore, patients can be analyzed as an aligned cohort in the block view and their temporal patterns can be viewed by switching to the timeline view.

6.3.4 Feature Operations

`OncoThreads` supports a wide range of data types, including gene-specific data like mutations or expression as well as clinical data, which may be timepoint- and patient-specific or just patient-specific.

Clinical data is pre-loaded upon study selection, while gene-specific data is queried on-demand using the HUGO gene symbol and the datatype of interest. `OncoThreads` includes a Feature Manager to transform features and change the order of currently displayed features. Additionally, a Feature Explorer is provided for the discovery of features to be added to the visualization via guided exploration [171]. For convenience, known features of interest can also be selected using a drop down menu in the toolbar of the visualization.

Feature Manager

Features are added to the view exactly as the data is provided, which may not be optimal for visualization. For example, application of a log scale might enhance the interpretation of continuous data with a wide range of values or combining multiple genetic features can enable pathway-based analysis. Therefore, the Feature Manager enables users to transform features (Figure 6.1d,e). All currently displayed features can be modified. Continuous features can be log transformed or binned to transform them to an ordinal feature, categorical features can be converted to ordinal features and vice versa, and binary features can be inverted. Moreover, features of the same kind can be combined. For example, binary features encoding for the presence of mutations in specific genes can be combined using a Boolean operator in order to quickly identify patients or groups of patients showing a combination of these mutations. In addition, every feature can be renamed and the color scale can be changed. The Feature Manager also enables changing the order of the features in the view, either manually or through sorting by an attribute like datatype, source (clinical, expression, mutation etc.), or name. In Section 6.4 we demonstrate the usage of the feature operations in a case study.

6.3.5 Feature Explorer

The Feature Explorer supports guided exploration and selection of features (Figure 6.1 c). It provides an overview of all clinical features and any genomic or molecular features that have been added, including range for continuous features, or data types, data source etc. In addition, the Feature Explorer provides variability scores

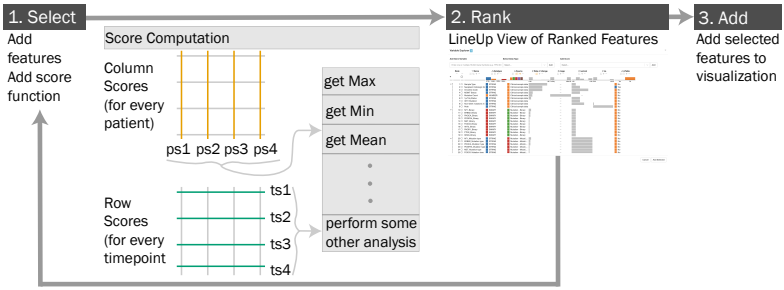


Figure 6.3: The framework shows four basic steps for feature exploration (i) Column scores and row scores can be selected to assess both variability within timepoints and across timepoints for each feature in the Feature Explorer. Scores are calculated for each timepoint or patient and aggregated using a method of choice (grey box). (ii) Features can be ranked by the calculated scores using LineUp [180]. (iii) Features of interest can be selected in LineUp and added to the visualization.

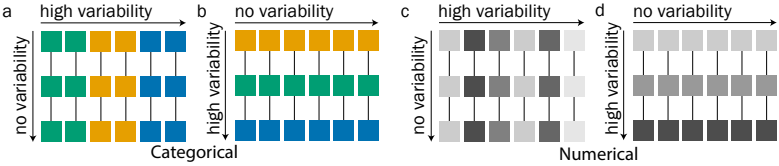


Figure 6.4: The examples in (a) and (c) show high variability within a timepoint, but no variability across timepoints for categorical and numerical data. Examples in (b) and (d) show the opposite pattern.

to highlight features that may be of biological interest due to high variability within a timepoint or across timepoints. These scores are measures of statistical dispersion that indicate the extent to which a distribution is stretched or squeezed. Users can select different scores using a drop down menu and can see the ranking of every feature based on these scores (Figure 6.3). This ranking is shown with an interactive technique called `LineUp` [180] which helps users prioritize features, evaluate them, and understand any correlations among them. Similarly to `StratomeX` [154], [171], features can be selected in `LineUp` and added to the visualization.

We examine two types of variability of features in `OncoThreads`: within timepoint and across timepoints (Figure 6.4). Variability within a timepoint examines how consistent the data for a feature is across all patients at each timepoint. Variability across timepoints examines how a feature changes over time for individual patients. Figure 6.4a shows data with high within timepoint variability, but low variability across timepoints. In contrast, Figure 6.4b shows low variability within timepoints and high variability across timepoints. A similar concept can be applied for numerical data (Figure 6.4c,d). However, different methods are required to calculate variability scores for the different data types.

Variability scores can be calculated both within timepoints (row scores) and across timepoints (column scores). We can aggregate these scores to obtain a single score for every feature. For example, consider a feature in four timepoints. We can calculate variability scores for this feature for each timepoint. These scores can then be aggregated to a single score by selecting the maximum, minimum or average of the four timepoint scores (Figure 6.3). Scores for all features can be compared within the Feature Explorer, allowing a user to rank features and find correlations among them. ModVR measures variation around the mode [181]. It is a standardized form of the variation ratio, a measure of statistical dispersion in nominal data, or the proportion of cases which are not in the "mode" category. The ModVR values range from 0, indicating low variability, to 1, indicating high variability. The Coefficient of Unalikeability measures variability for categorical data. It represents the proportion of observations that differ. The higher the value, the more unalike the data are [182]. The Coefficient of Variation (CV) is the ratio of the standard deviation to the mean. A CV less than 1 indicates low variance, whereas a CV greater than 1 indicates high variance. For categorical features the Rate of Change is the number of values that changed relative to the total number of value transitions. For continuous features it represents the rate of the average change to the observed range. Developers can implement additional scores for this extensible ranking framework.

Design Process

We employed the design sprint methodology [179] to enable our multi-institutional team to develop consensus goals as well as to obtain user feedback prior to undertaking a full development and implementation process. We also set out to evaluate the success of applying a design sprint to visualization problems. We performed the design sprint with a group of six people with backgrounds in biology, biomedical informatics, and visualization over five consecutive days, for 6 hours each day. The overall goal for our effort was to "develop the 'go-to' visualization approach for longitudinal cancer molecular data through an agile framework that will have measurable technical and scientific impact."

As part of the process, we interviewed three cancer researchers for 30 minutes each in addition to the authors to identify the most important challenges that needed to be addressed, which raised questions such as: "How might we visualize an entire cohort over time?"; "How might we integrate multiple data types into one visualization?"; "How might we define timepoints?", and "How might we enable the flexible analysis of a cohort relative to any event, e.g. diagnosis or treatment?"

We examined existing tools and visualization strategies, including StratomeX [154], Domino [178], streamgraphs and Sankey diagrams; these inspired sketches from which we decided to utilize heatmaps and Sankey diagrams as the core components of the visualization. The visualization consists of connected blocks with the rows representing multiple features at different points in time. In order to facilitate finding patterns in the data, users can switch between the heatmap and the Sankey diagram as well as sort the visualization by a chosen feature. We reviewed an existing cancer evolution study [183] and used one of its main findings to define a path through the data which we could implement as a prototype of linked slides with the presentation software Apple Keynote. Given the time constraints of the design sprint, the prototype allowed for just a single path, rather than all possible paths of exploration.

We tested the prototype with four cancer researchers, all of whom successfully arrived at the scientific conclusion that we intended and

found the tool useful overall. However, users also identified many opportunities for improvement; the primary issues were that users struggled due to the limited interactivity of the prototype and that the Sankey visualization in the prototype was confusing and did not provide an advantage over the heatmap.

Based on the feedback we received, we made two major changes to the concept: (i) instead of sorting the whole visualization by a single feature, we enabled independent sorting for each block, and similarly (ii) transform from a heatmap to a Sankey diagram iteratively by grouping blocks individually. The independent sorting and grouping of blocks prevents the visualization from changing too quickly, which we identified as a potential reason for misinterpretation of the prototype visualization. Moreover, selectively viewing blocks in detail enhances the exploration by adding flexibility.

6.3.6 Availability and Implementation

`OncoThreads` is a web application available at `http://oncothreads.gehlenborglab.org` and its source code is available at `https://github.com/hms-dbmi/oncothreads` under the MIT license. `OncoThreads` is implemented in JavaScript using the libraries React (`https://reactjs.org/`), mobx (`https://mobx.js.org`) and D3 (`https://d3js.org/`) [146] for the application structure, state management, and visualization respectively. React-bootstrap (`https://react-bootstrap.github.io`) has been used to apply bootstrap styles to the React components. We retrieve data from the cBioPortal using their REST (Representational State Transfer) API with the promise-based library axios (`https://github.com/axios/axios`). Additionally, `OncoThreads` can be obtained as an Electron app (`https://electronjs.org`) available for download at `https://github.com/hms-dbmi/oncothreads/releases`.

6.4 Results: case study in low-grade glioma cohort

In a study by Johnson et al., the authors explored the genomic evolution of low-grade glioma by analyzing a cohort of 23 patients

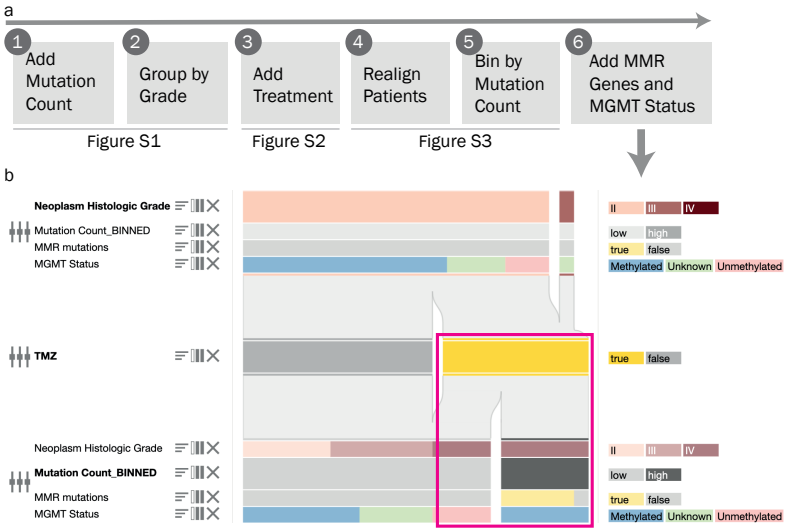


Figure 6.5: Exploring the glioma data set of a study by Johnson et al.. (a) Overview of the exploration. (b) Detailed view of the final step. Time is vertical, and two timepoints are shown. We can observe that most patients are classified as grade II at timepoint 1, and that most patients progress to grade III or IV at timepoint 2. Further, all patients with a high mutation count received prior TMZ treatment and methylation of MGMT and mutations in mismatch repair genes co-occur with high mutation count (magenta box).

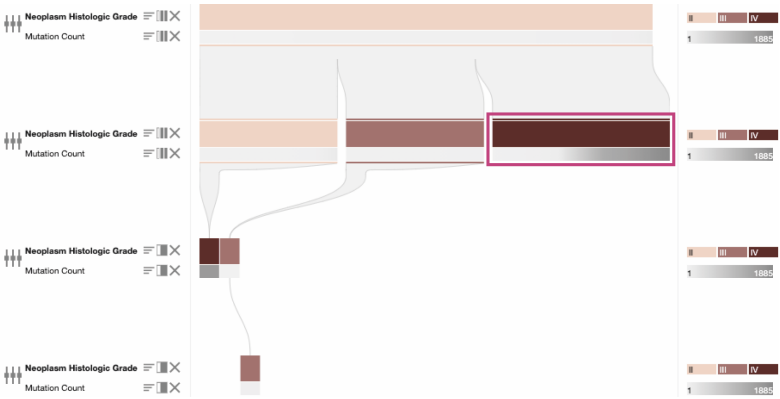


Figure 6.6: First step of example analysis of glioma data using OncoThreads. Four time points are displayed vertically, along with two features at each time point: Neoplasm Histologic Grade and overall mutation count. Timepoints 1 and 2 are grouped by Neoplasm Histologic Grade. Visual inspection shows that all patients are grade II at timepoint 1, but many develop a higher grade tumor at later timepoints. Grade IV tumors at timepoint 2 are also associated with increased mutation rate (highlighted in magenta outline).

with samples from an initial resection as well as one or more recurrences [183]. Samples were profiled with whole-exome sequencing and patients were clinically annotated. Among the findings of the paper was the impact of the chemotherapy temozolomide (TMZ) on low-grade gliomas; in six patients, tumor samples acquired after treatment with TMZ showed hypermutation and progression to high-grade glioblastoma in the context of MGMT silencing and loss of mismatch repair.

Figure 6.5a illustrates specific steps in an exploration of the data from Johnson et al. that demonstrates how the features of `OncoThreads` support the discovery of relevant subgroups within the patient cohort. After selection of the relevant data set ("Low-Grade Gliomas (UCSF, Science 2014)"), a single feature, neoplasm histologic grade, is automatically rendered in the block view. By using the Feature Explorer and applying the Rate of Change score, we

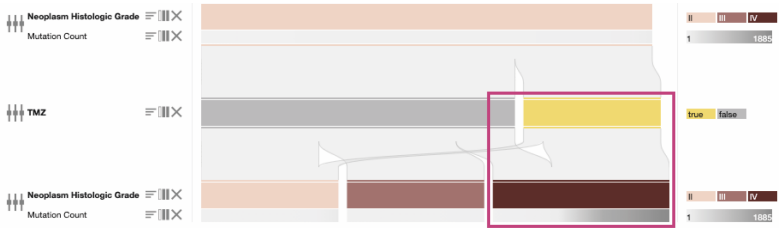


Figure 6.7: Second step of an example analysis of glioma data using OncoThreads. TMZ treatment has been added and the event block between timepoints 1 and 2 has been grouped by TMZ. Most patients showing a high grade have been treated with TMZ (highlighted in magenta outline).

find that several features, including mutation count, show variability over time and are therefore especially interesting for analyzing differences between initial resection and recurrence. To explore the temporal patterns in more detail, we add mutation count and group both timepoints 1 and 2 by neoplasm histologic grade. This allows us to visualize specific trends in the data; for example, we observe that all patients have grade II tumors in the first timepoint block, but many develop a higher grade tumor at later timepoints. We also observe significantly increased mutation counts in grade IV tumors at timepoint 2 (Figure 6.6). We can now ask what factors may have influenced tumor development from grade II to grades III and IV.

In the Feature Manager we add temozolomide treatment (TMZ), and subsequently group the event block between timepoints 1 and 2 by TMZ treatment. We can then see that there is a notable flow from patients receiving TMZ to patients having a high grade in the second sample, suggesting that TMZ treatment may result in a higher grade recurrence (Figure 6.7). To further assess the effect of TMZ treatments for all patients, we realign the entire cohort relative to the treatment. We also want to see if the patients who received TMZ and developed a high grade recurrence also have a high mutation count. Since mutation count is a continuous feature, we have to bin it first to transform it into a categorical feature as described in Section 6.3.4. Based on the distribution of mutations indicating that there are six samples exhibiting very high mutation counts, we create bins for low ($<$ 150 mutations) and high ($>=$

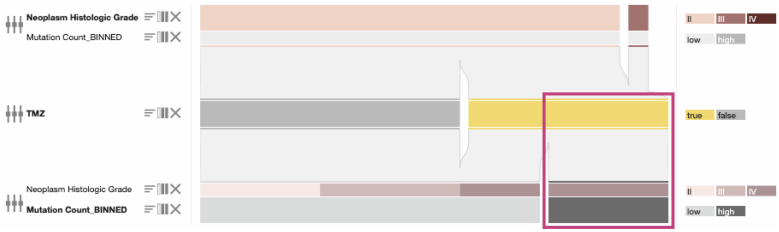


Figure 6.8: Third step of an example analysis of glioma data using OncoThreads. Patients have been realigned based on TMZ treatment, and mutation count has been binned into high and low. All patients with a high mutation count were treated with TMZ and have a grade IV tumor (highlighted in magenta outline).

150 mutations) mutation counts. Based on this exploration we can formulate the hypothesis that TMZ treatments correlate with high mutation count and grade IV at recurrence (Figure 6.8).

Given this correlation between TMZ treatment and increased mutational burden, we next look for additional evidence to functionally connect these two features. TMZ is a mutagen, and TMZ-induced mutations are believed to be mitigated by MGMT protein and the mismatch repair pathway [184]. Leveraging the available molecular data, we add additional tracks to show the mutational status of mismatch repair pathway genes MLH1, MSH6 and MSH3, and then use the Feature Manager to combine those tracks into a single track showing the overall mismatch repair pathway mutation status. We also add a track showing the MGMT methylation status of each sample. Now, examining those samples with high mutation count following TMZ treatment, we see that all samples show methylation of MGMT, indicating silencing of the gene and subsequent lack of protein, and almost all have mutations in mismatch repair pathway genes, which together support a potential causative role for TMZ in inducing hypermutation in these tumors (Figure 6.5b).

6.5 Discussion

6.5.1 Application

The results of the case study demonstrate how the visual exploration features of `OncoThreads` support users in efficiently generating testable hypotheses and identifying supporting evidence through an effective combination of visualization and data integration tools. For example, `OncoThreads` helps researchers to explore the influence of a specific treatment on tumors in an entire patient cohort and to find patterns for the prediction of the outcome of a therapy. Furthermore, it may be used to discover patterns of genetic predispositions which can affect the effectiveness of a drug or help analyze the effects of different drug dosages.

Currently, `OncoThreads` utilizes variation around the mode (ModVR) for categorical data, and variance or coefficient of variation for numerical data [185] to rank features based on variability (Figure 1c). However, these variability scores are implemented in an easily extensible framework, such that additional scores or aggregation approaches can be added, for example, calculating the variability score of a single timepoint rather than the aggregate across all timepoints to enable a query like "How do the features compare to each other based on their variability in timepoint 2?"

In the future, additional user interactions could trigger more complex queries in `OncoThreads`. An example of such a query could be: "Find all features that show a similar pattern in a specific timepoint". Such a query would help users to identify correlations among features. In addition to queries involving sample features, event features could be taken into account in scoring functions to evaluate their relationship to sample features of subsequent timepoints. In general, these scoring mechanisms could guide users to features that provide additional insights and to generate new hypotheses.

With the undo and redo operation `OncoThreads` allows going back to previous steps during the exploration process. Yet, when a new action is performed after undoing, the previous path of exploration is lost. Therefore, it would be desirable to incorporate visualization provenance approaches such as `Vistories` [186] or `Trrack` [187]

into OncoThreads. In those approaches the user's actions are saved in a graph that captures all relevant interactions. Therefore, it is possible to go back to parts of the exploration that would be lost in regular undo/redo implementations. Moreover, those approaches allow the presentation of the results of the exploration by enabling the creation of a "replay" that communicates the results by showing certain steps of the exploration with annotations.

In the future, we plan to improve scalability in the number of features and timepoints. One promising direction is to integrate sequential pattern mining and clustering techniques into the visual exploration of longitudinal patient data. These techniques can effectively learn patterns from complex sequential data and facilitate the identification of disease states. Moreover, we plan to enhance the representation of patient specific data, as well as tumor heterogeneity. Although OncoThreads has been developed specifically for cancer data, it can also be applied to many other kinds of multidimensional temporal data.

6.5.2 Design Sprint

To our knowledge, the design sprint technique has not been documented for the development of a biomedical data visualization tool before. In the examples described by Knapp et al., the design sprint methodology is used for the development of tools and products without exploratory functionality [179]. For example, when a website is designed for selling a product there are a few well defined steps that a user has to conduct to purchase the product. In contrast, an explorative visualization can be used in many different ways and no clear endpoint is defined. Therefore, we recommend adapting the technique to visualization problems, especially to deal with the complexity of modeling their exploratory and interactive nature. For example, defining the workflow of the planned tool before conducting user interviews might introduce bias in the downstream process. It might be more useful to define required steps without specifying their order. Moreover, during prototyping, it is likely not feasible to implement all possible exploratory steps in the given timeframe, so we had to limit the exploration to one path. Similarly, time for sketching needs to be increased. Nevertheless, we found that the

approach can be applied effectively to efficiently develop and test ideas despite the complexity of the data and to create a shared vision for the team. While the design sprint technique allowed us to get early feedback from users, a validation of `OncoThreads` with an insight-based evaluation approach [188] could provide more information about the quality of the hypotheses generated with `OncoThreads`.

Chapter 7

Visualizing Networks: ProtEGOnist

Small-world networks, such as protein-protein interaction networks or gene co-ocurrence networks, are common in biology. The central property of small-world networks is the high clustering coefficient and low distances. This means that the shortest path between any two nodes is comparatively short as the network contains central nodes with many interactions. Visualizing small-world networks as simple node-link diagrams often leads to hairballs, which led to the creation of the Bio+MedVis Challenge 2023. This challenge asked for contributions comprehensively visualizing a large Protein-protein interaction network while being able to also focus on single proteins of interest and their interactions. The approach `ProtEGOnist` presented in this chapter was the winner of this challenge. It solves this challenge by aggregating the network into ego-graphs, which reduces its size and provides a structured network layout.

This chapter includes previously published work on `ProtEGOnist`, presented at the EuroVis Conference 2024 [29]. For readability, funding information has been removed and figures originally included in the supplementary material have been included in the main text. Supplementary Figure S2, showing a larger cutout of the table displayed in Figure 7.7 has been removed.

ProtEGOnist: Visual Analysis of Interactions in Small World Networks Using Ego-graphs

7.1 Abstract

Visualizing small-world networks such as protein-protein interaction networks or social networks often leads to visual clutter and limited interpretability. To overcome these problems, we present

ProtEGOnist, a visualization approach designed to explore small-world networks. ProtEGOnist visualizes networks using ego-graphs that represent local neighborhoods. Ego-graphs are visualized in an aggregated state as a glyph where the size encodes the size of the neighborhood and in a detailed version where the original network nodes can be explored. The ego-graphs are arranged in an ego-graph network, where edges encode similarity using the Jaccard index. Our design aims to reduce visual complexity and clutter while enabling detailed exploration and facilitating the discovery of meaningful patterns. To achieve this, our approach offers a network overview using ego-graphs, a radar chart for a one-to-many ego-graph comparison and meta-data integration, and detailed ego-graph subnetworks for interactive exploration. We demonstrate the applicability of our approach on a co-author network and two different protein-protein interaction networks. A web-based prototype of ProtEGOnist can be accessed online at https://protegonist-tuevis.cs.uni-tuebingen.de/.

7.2 Introduction

Networks are used to model a wide array of systems. Depending on the underlying data, networks can differ in their parameters, size, density, and connectivity. Many networks such as social networks, biological networks, transportation networks, or citation networks exhibit the *small-world property*, which means that most nodes can be reached from any other node in a small number of steps [189], [190]. A famous example of this property is the idea of the 6-handshakes rule also known as *six degrees of separation*, stating that every person in the world only has a distance of a maximum of six handshakes from any other person [191], [192].

The small world property is also prevalent in many biological networks [193] like *protein-protein interaction* (PPI) networks, which play a crucial role in modeling and understanding the intricate mechanisms governing cellular processes. In PPIs, proteins are seen as nodes, while interactions are represented by edges. Traditionally, two proteins are considered interacting if they bind physically.

However, the concept is often extended to other indirect connections, such as the spatial proximity of the corresponding genes in the genome or co-occurrence validated in the literature.

Typical visualizations for small-world networks include node-link diagrams or matrix representations [194]. Links in our case represent any type of interaction, e.g., interacting proteins or "interacting" researchers co-authoring a paper. Visualizing complete networks with many thousands of nodes as node-link diagrams typically results in a cluttered, hairball-like structure, especially when using standard force layouts [195]. Moreover, the sheer number of nodes and interactions makes it challenging to find nodes of interest and do a targeted comparison of their neighborhoods.

Oftentimes, single nodes serve as starting points when analyzing small-world networks, such as oneself or a famous person in a social network. Usually, it is meaningful to inspect not only immediate contacts but also indirect ones. Social science studies have shown that in social networks such indirect contacts can affect, e.g., a person's happiness [196] and their ability to find a job [197]. In PPI networks, a node of interest could be a protein that is the research focus of a biologist. Indirect connections are studied, e.g., when analyzing metabolic pathways. This is, e.g., important in PPIs showing physical interactions. It has been shown that proteins with the same interaction partners rarely interact directly [198]. Common path lengths for PPI networks are between four and five [199], thus, contacts with a distance higher than two tend to cover very large portions of the network [200].

To study such local subnetworks around nodes of interest, ego-graphs can be used [201]. Originally developed for the study of social networks, this approach focuses on the local neighborhood of an individual node, instead of showing all nodes and interactions. An ego-graph consists of a central node of interest—the *ego*—and its local neighborhood in the network—the *alters*. Degree-1 alters are alters with a direct connection to the ego. Degree-2 alters have direct connections to degree-1 alters, but not to the ego. That is, 1-level ego-graphs only consider degree-1 alters, and 2-level ego-graphs consider degree-2 alters as well. Typically, 2-level ego-graphs are used [202], i.e., "friends-of-friends" networks.

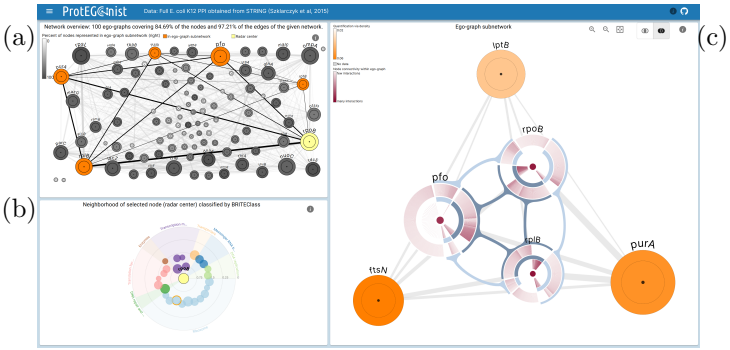


Figure 7.1: `ProtEGOnist` uses ego-graphs for the visualization of small-world networks (in this case, the *Escherichia coli* protein-protein interaction network). (a) The network of ego-graphs visualized as aggregated glyphs gives an overview of the network with edges visualizing similarity. (b) A radar chart shows the similarity between a selected ego-graph (center, yellow node in (a)) and its neighbor ego-graphs grouped by metadata. Circles representing ego-graphs that are included in the subnetwork are highlighted by an orange outline. (c) An ego-graph subnetwork selected from the overview (orange nodes in (a)) provides details. Three ego-graph nodes are deaggregated forming an ego-graph group for a detailed comparison. The ego-graphs contain all nodes with a maximum distance of two.

We developed `ProtEGOnist`, a novel visualization approach for the exploration of small-world networks that uses 2-level ego-graphs to aggregate local neighborhoods represented by glyphs. (Figure 7.1). Initially, `ProtEGOnist` was submitted as a contribution to the Bio+MedVis Challenge 2023 [203] for redesigning the visualization of a specific PPI network by Gonçalves *et al.* [204]. The challenge data set included a PPI network together with protein-drug associations predicted using a deep learning approach developed by the authors called *DeeProM*. The original visualization was a static figure (Figure 7F in original publication) showing the PPI network as a node-link diagram with 8,395 nodes and 66,721 edges. The main point of the challenge was to provide a less cluttered view of the network that includes overviews as well as detailed visualizations that are enriched with metadata. Moreover, the challenge specified the need to focus on specific proteins and explore their relationship. Based on the visualization of this data set, `ProtEGOnist` was awarded as the best contribution to the Bio+MedVis Challenge 2023.

Our contributions can be summarized as follows: we present a universal approach for the exploration of small-world networks with many thousands of nodes. Our approach focuses on ego-graphs, i.e., placing the individual in the center as the ego and thus making it the "protagonist" of the graph (subsection 7.4.1). To achieve this, `ProtEGOnist` creates a network of ego-graphs from an input interaction network and uses specifically designed glyphs to visualize these ego-graphs (subsection 7.4.2). The edges in the network of ego-graphs are weighted by the similarity of the ego-graphs, which is calculated as the Jaccard index of the node sets for each pair of nodes. This concept allows for an exploration from an overview level down to analyzing subsets of ego-graphs, comparing up to three ego-graphs in detail, and inspecting single ego-graphs (subsection 7.4.3). Using the taxonomy proposed by Filipov *et al.* [194], we would position `ProtEGOnist` as a *group network visualization*, since the ego-graphs represent groups of nodes based on neighborhood. Using the *Vertex Group Structure Taxonomy* by Vehlow *et al.* [205], we would describe `ProtEGOnist` as an *overlapping hierarchical* structure, which they found only in a single approach.

We demonstrate the effectiveness of our approach using three exemplary use cases. The first one is a co-author network built from IEEE VIS authors [206], i.e., a social network (subsection 7.5.1). The other two show that our approach can be used for domain-specific data sets, such as visualizing PPI networks: we applied `ProtEGOnist` to a PPI network of *Escherichia coli* (subsection 7.5.2) and a human PPI with metadata on drug-protein associations provided for the Bio+MedVis Challenge 2023 (subsection 7.5.3).

7.3 Related Work

Simple node-link diagrams are the most commonly used visualization techniques for networks [207]. They are, e.g., popular for visualizing PPIs and are used by `STRING` [22] and the well-known network visualization tool `Cytoscape` [208]. Although node-link diagrams are conceptually intuitive and powerful for visual analysis, they quickly suffer from overdraw, layout problems, and clutter for larger networks [195].

Therefore, approaches beyond force-directed node-link diagrams, different layouts [209] including hierarchical layouts [210], on-node encodings [211], and hybrid network visualizations [212], [213] have been developed. As an in-depth review of the vast literature on network visualization is beyond the scope of our paper, we refer to the recent state-of-the-art reports by Filipov *et al.* [194] and Nobre *et al.* [207].

To reduce the amount of clutter created by the edges, various approaches are used. Edges can be partially indicated [214] or even omitted completely. An omission is only viable if edges are implicit or not of interest in the applied layout, for example, when applying containment to cliques. Both approaches can be enriched by drawing the edges in full on-demand [215]. Moreover, edge bundling can be used to use topological or semantic information to merge edges into bundles [216], [217]

A general approach to reduce the clutter in a node-link diagram is to reduce the number of elements via grouping, clustering, or aggregation. Grouping can be utilized using underlying semantic

information to generate containment for groups sharing specific attributes [215], [218]. These approaches depend on semantic metadata suitable to the applied grouping method and the underlying goal of the analysis. Alternatively, networks can be grouped purely by topological measures, for example, grouping into subnetworks of densely connected nodes or by creating ego-graphs for a set of manually or computationally determined nodes of interest.

Aggregation is often used by merging nodes into distinct glyphs, increasing readability [219]. In one such example, Vehlow *et al.* [220] visualize multiple overlapping hierarchical networks using node-link diagrams. In their `fuzzy-communities` approach, they display an overview using multiple levels of abstraction. Depending on the chosen level, some or all nodes are collapsed into meta-nodes, which encode network membership-heterogenicity using the fuzziness of the shape.

Alternative techniques go further by substituting glyphs for other standalone visualization types, like adjacency-matrices [212], chord diagrams [213] or customizable plots such as line- and bar-charts [211]. While these approaches aim at visualizing groups in general, specialized visualization types have been developed for ego-graphs, aiming at displaying their inherent hierarchical structure with an ego and alters. The `EgoComp` approach uses a hybrid network visualization for comparing ego-graphs in social networks [221]. It applies both an implicit hierarchical layout for the visualization of ego-graphs and a conventional node-link layout for linking identical nodes between the compared graphs. For the visualization of ego-graphs, nodes are placed around a center in partial circles according to their distance from the ego. The half-circles of the two compared ego-graphs are facing each other. Since two ego-graphs can contain the same nodes, edges connect the respective nodes to express identity.

Ego-graph visualizations are extensively used in the domain of dynamic graphs, as shown in a recent review by Kale *et al.* [222]. While dynamic graphs represent a data structure with specific tasks and use cases, some of the visualization concepts apply to static graphs. Visualizations of ego-graphs in dynamic networks include node-link diagrams using a stress-majorization layout [223] circular

glyphs [224], radial layouts [225], [226], linear layouts [202], [227] as well as combinations of the aforementioned approaches [228]. `EgoSlider` aggregates an ego-graph into a pie- or bar-chart glyph. They encode changes in the properties of the graph at different time points. The authors use the circular variant as default, arguing that the design serves as a "metaphor of an ego surrounded by alters" [224]. This metaphor corresponds to a radial approach to visualizing target-based graphs, where the center node of the node-link diagram effectively considers the graph as the root of a rooted tree. It is an approach commonly found in the literature [229]–[231], especially in literature focused on ego-graph visualization [221], [232], [233]. Differently, `EgoLines` is an example of a technique applying non-circular layouts when comparing an ego-graph at different states of a dynamic network [202]. It visualizes ego-graphs as adjacency matrices, but as they grow quadratically in size with an increasing number of nodes, they are hard to interpret for large ego-graphs. The ego is placed in a central position with alters placed outwards, inducing a hierarchy within the alters even if not desired.

In addition to the general approaches, we also consider ego-graph approaches for PPI networks relevant to our work as a contribution to the Bio+MedVis Challenge [203]. The `STRING` database [30] is a popular resource for PPI networks. It uses different interaction types to calculate a confidence score for each protein-protein interaction. Ego-graphs are shown when searching for a protein of interest [22]. The alters, are the proteins that are directly connected with a query protein – the ego. This 1-level ego-graph is shown as a node-link diagram with a force-directed layout. By default, it displays only the 10 highest-scoring interactions to reduce the network size. Optionally, a second shell of interactions can be displayed, showing the highest-scoring direct neighbors of the interactors of the target query (2-level ego-graph). In contrast to `STRING`, `BioLinker` [234] visualizes the entire network in an overview, where nodes of interest can be selected and visualized as a subnetwork consisting of ego-networks in a separate view. Moreover, `BioLinker` highlights the egos such that they are visually distinct from the alters.

Another application using ego-graphs in biological networks is the `EgoNet` algorithm [235], which identifies disease subnetworks.

EgoNet can be applied to PPIs where each protein is associated with protein abundances (also called *protein expression*) at different clinical outcomes, e.g., when comparing protein abundances in healthy and cancerous cells. Starting with an ego, the tool iteratively adds alters and calculates if the contained proteins suffice to predict the clinical outcome. This approach shows how ego-graphs are used as a data structure to computationally reduce the network size by focusing only on the most relevant nodes.

7.4 Approach

Based on the Bio+MedVis Challenge 2023 [203] and aided by the task taxonomy for graph visualization by Lee *et al.* [236], we identified the following tasks for the development of ProtEGOnist:

Overviews can provide a starting point for the analysis [95], especially in previously unexplored data sets and when there is no clear hypothesis about the data. For this, we want to simplify the network and declutter it by aggregating groups of nodes into meta-nodes. Then we can exploit the small-world property to facilitate an overview showing the important meta-nodes, like those covering a large part of the interaction network. This can be described as an *Overview Task*.

Often the global context of entities of interest identified before exploration, such as specific known individuals within a social network or proteins within a PPI is of special interest. Van Ham and Perer [237] present an approach applying the *"Search, Show Context, Expand on Demand"* principle, which focuses on nodes of interest that can be interactively added to the visualization and shown in the context of the graph. A central aspect of ProtEGOnist should be the selection of nodes-of-interest, which is best described as an *Attribute-Based Task - On the Nodes*. For this, we want to create the meta-nodes based on the neighborhoods of nodes of interest and need to find the nodes accessible from these nodes (*Topology-Based Task - Accessibility*).

Moreover, we want to empower users to find similarities between nodes, for example, by estimating the overlap between meta-nodes.

We also want to allow for a more meaningful and in-depth comparison of meta-nodes. A user might want to find the nodes shared between the corresponding neighborhoods. Both actions correspond to a *Topology-Based Task - Common Connection*. Finally, we also want to allow the users to utilize the metadata layer to find nodes fulfilling domain-specific criteria. For example, metadata should be used for filtering nodes of interest and mapped to visual channels. This can again be described as an *Attribute Task - On the Nodes*.

Based on these described tasks, we identified the following requirements for `ProtEGOnist`:

R1 Overview: Apply filtering and aggregation techniques to provide a comprehensive overview showing the most relevant meta-nodes(e.g., representing numerous interactions).

R2 Subnetwork context: Viewing meta-nodes in the local and global network context.

R3 Detail: Allow a detailed analysis of meta-nodes, such as finding shared nodes in subnetworks.

R4 Metadata: Provide the integration of further metadata on the network, such as categories or measurements for the instances represented by nodes.

We want to develop a layout that satisfies these requirements and enables the defined tasks and thus results in a less cluttered visualization in comparison to force-directed node-link diagrams like the one of the Bio+MedVis Challenge.

7.4.1 Ego-graph Concept & Visualization Design

We address the requirements defined above with `ProtEGOnist` using ego-graphs. Interaction networks consist of nodes representing entities, such as proteins in PPI networks or people in social networks, and edges representing interactions between them. Instead of visualizing every node and interaction individually, `ProtEGOnist` groups nodes and interactions into ego-graphs and represents them as circular glyphs. Similarity values are calculated for every pair of ego-graphs using the Jaccard index of the sets of contained nodes,

i.e., the intersection size divided by the size of the union. Using the similarity values, an *ego-graph network* is created, where the nodes are visualized using the ego-graph glyphs. The small world property can be exploited to create an overview since a comparatively small set of ego-graphs is sufficient to cover a relatively large set of nodes of the original network (**R1**). The glyphs provide additional details about the ego-graphs contained in the network and their relation to each other (**R2**,**R3**).

We use 2-level ego-graphs, i.e., all alters have at most a distance of two to the ego, to achieve a reasonable reduction of the original network as well as to offer visually feasible comparisons between any two ego-graphs. Each node can be chosen as the ego of an ego-graph. To represent a single ego-graph, we have designed two types of radial glyphs: a detailed one and an aggregated one (Figure 7.2a). The detailed glyph (Figure 7.2a, top) visualizes the alters as ring segments in two circular levels around the ego (**R3**). The circular layout highlights the central role of the ego and represents a space-efficient layout for its alters. The first, inner level contains all degree-1 alters, while the second, outer level contains all degree-2 alters. The ego is represented by a filled circle in the center. To visualize the connectedness of the alters, the ring segments representing the alters as well as the ego circle are colored to represent their node-degree in the network (*few interactions*  *many interactions*). To avoid clutter, the *interaction edges* of the alters are only shown on hover. The aggregated glyph (Figure 7.2a, bottom) is a simplified, abstract version of the detailed glyph. It consists of two concentric circles to symbolize the two levels and a black dot in the center to represent the ego. The background of the glyph can be colored to represent a certain property of the underlying ego-graph. The size of both the detailed and the aggregated glyph can be scaled to illustrate the number of elements in the ego-graph, that is, the cardinality of the ego-graph itself. For the detailed glyph, this is a deliberate double encoding that makes it easier to compare the size of two ego-graphs instead of counting the circle segments representing the nodes. Optionally, circular text labels on top of the glyph can show the name of the ego (Figure 7.1). The font size is scaled with the size of the glyph, and the text labels are automatically discarded if the glyph is too small.

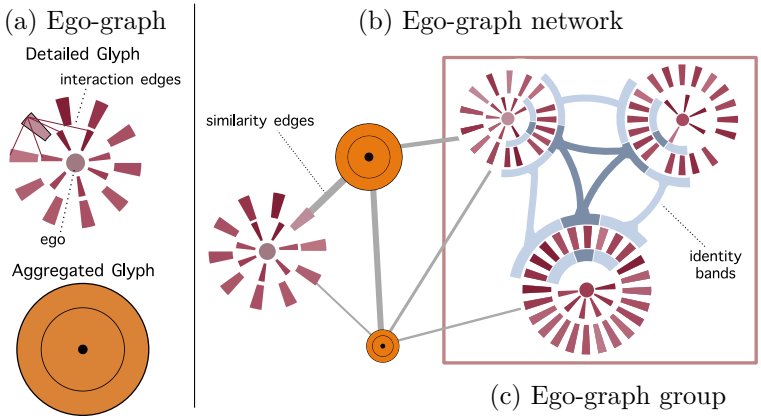


Figure 7.2: Ego-graphs and ego-graph networks, the concept of `ProtEGOnist`. (a) A single ego-graph can be visualized in detail or aggregated. The detailed view shows the ego in the middle of a circular graph layout. Degree-1 alters are placed on the inner circle, and degree-2 alters on the outer circle. Nodes are colored corresponding to their number of interaction edges (*few interactions* *many interactions*). The interaction edges are only displayed when hovering over a node. The aggregated view of an ego-graph encodes the number of alters via the area of the glyph. (b) An ego-graph subnetwork consists of single ego-graphs and ego-graph groups. In this view, similarity edges connect single ego-graphs. Their width and opacity encode the Jaccard index between the respective ego-graphs. (c) Visualization of an ego-graph group. Ego-graph groups are arrangements within the ego-graph network of up to three detailed ego-graphs with *identity bands* connecting identical nodes. A darker blue indicates nodes occurring in all three ego-graphs, while the lighter blue indicates only pairwise intersections.

The glyphs can also be used as the nodes of an ego-graph network (**R2**), where the size of each node encodes the cardinality of the ego-graph and the edge widths encode the similarity using the Jaccard index (*similarity edges*, Figure 7.2b). By default, ego-graphs are represented as aggregated glyphs that can be expanded on demand to show the detailed glyph for an in-depth analysis.

Connected ego-graphs can be selected to form an *ego-graph group* to show in detail which alters are shared between the ego-graphs or unique to an ego-graph (**R3**). The groups show the ego-graphs as detailed glyphs and the numbers of shared alters as *identity bands* (Figure 7.2c). We restrict groups to three ego-graphs to eliminate crossing bands. In the case of an ego-graph group with three ego-graphs, we divide each ego-graph circle into four sections: one for alters unique to the respective ego, two for alters shared between any two ego-graphs, and one for alters shared between all three ego-graphs. The three detailed ego-graph glyphs are placed to form an equilateral triangle. The sections shared between all three graphs are arranged to face towards the center of the imaginary triangle (dark blue), while the pairwise sections face towards each other (light blue), and the section containing the unique nodes faces away from the triangle center. The shared sections are illustrated by contour arcs covering the corresponding nodes of the detail glyphs. The arcs on the glyph surfaces are connected via *identity bands* to visualize that the corresponding sections in the ego-graphs contain the same nodes. These curved bands are optimized to avoid sharp angles or crossings by positioning them off-center to the corresponding arc. The colors of the bands match the arc color and facilitate distinguishing the portions of nodes shared between two and three ego-graphs. If the group only consists of two ego-graphs, only a single section is generated for the shared nodes, and the two ego-graphs are placed on a horizontal line.

The alters in the detailed ego-graphs are sorted separately. Within the sections, they are sorted by three criteria: (i) Their distance to the ego, (ii) their average distance to the other egos in the ego-graph group, and (iii) their node degree (**R2**). Thus, distinct subsections within the sections emerge, facilitating the location of shared nodes with a specific distance to the different egos.

7.4.2 Glyph and Ego-Graph Group Redesign

As an initial idea for the submission to the Bio+MedVis Challenge, we followed the approach implemented in *egoComp* [221], in which alters shared between two ego-graphs are connected using edges (Figure 7.3a). While this is feasible for comparing two ego-graphs in detail, we encountered several issues when using this in the ego-graph network and for ego-graph groups (**R3**).

To avoid edge crossings, the sort order of shared alters in ego-graph groups had to be identical in each graph. This caused some proportions of the ego-graphs to remain in a non-logical order concerning the node degree, and the general distribution of node degrees could not be deduced visually. Moreover, we could not use the entire circle for displaying alters shared between ego-graphs but only a portion to avoid edges crossing the nodes. In the case of an ego-graph group of size three, it was hard to visually distinguish alters that are shared between all ego-graphs and alters shared between only two ego-graphs. In addition, identity edges could not easily be distinguished from similarity edges. Furthermore, alters were encoded by circles in the previous version. For large ego-graphs, the available space to arrange alters around the ego is limited, leading to tiny radii when displaying circles. This in turn caused a very poor "ink-to-space" ratio, which then made it very hard to properly distinguish single nodes.

With the introduction of colored curved identity bands (Figure 7.3b), we addressed all of these issues. The usage of identity bands leads to the circle being split into sections, effectively creating a donut chart-like visualization of the grouping of nodes. Identity bands can be distinguished from the similarity edges through the colors and the organic shape. By drawing bands instead of individual edges, we can now use the entire circle to display shared nodes, allowing the creation of a second view mode that shows only nodes shared by any of the detailed ego-graph instead of all nodes for all ego-graphs (Figure 7.3c, *shared-only mode*). Lastly, the problem of a low "ink-to-space" ratio was tackled using ring segments instead of circles to visualize the alters, as explained in subsection 7.4.1. Furthermore, as individual identity edges are no

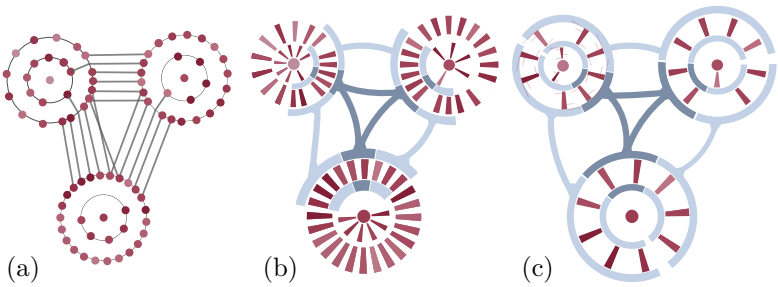


Figure 7.3: Different concepts for visualizing ego-graph groups: (a) First concept as submitted to the Bio+MedVis Challenge 2023. Nodes are encoded as circles and connected to the instances of the same node with identity edges. (b) Redesigned concept, where nodes are encoded as ring segments and identical instances are connected with identity bands. (c) Shared-only mode, where alters not shared with other ego-graphs in the group are filtered out.

longer drawn, more advanced sorting criteria could be introduced for the segments leading to a natural partitioning into subsections.

7.4.3 Visual Interface & Application Design

`ProtEGOnist` uses three main visualization components (Figure 7.1): a simplified *overview* of the original network showing a static ego-graph network (Figure 7.1a), a *radar chart* showing information about ego-graphs similar to one specific ego-graph (Figure 7.1b), and an *ego-graph subnetwork* (Figure 7.1c), which applies the concept of dynamically de-aggregating ego-graphs to a user-defined subset of the ego-graph network for a detailed analysis and comparison.

Overview

The *overview* shows a network of the most relevant ego-graphs (Figure 7.1a, **R1**). Depending on the data set, the set of most relevant ego-graphs is already known (section 7.5, DeeProM use case). For a general solution, we propose the following algorithm to extract an informative subnetwork of relevant nodes: provided that the input

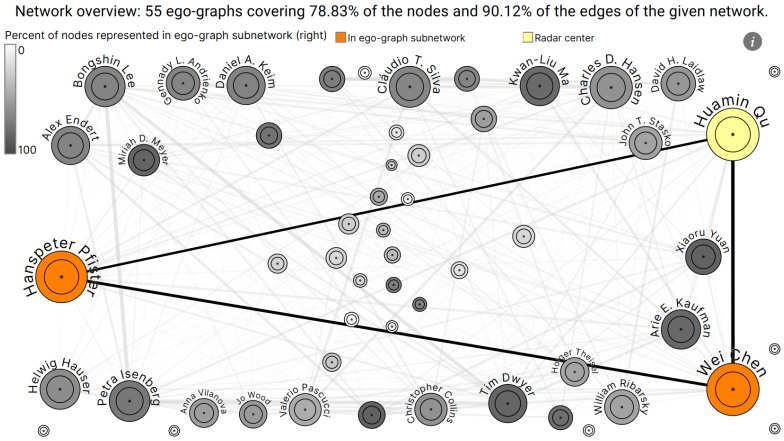


Figure 7.4: Network overview using a set of 100 ego-graphs representing authors in a co-authorship network, covering 83% of the nodes and 95% of the edges of the original network. The color scale from white to gray maps to the percentage of nodes in the ego-graph currently visualized in the ego-graph subnetwork (*0%* *100%*). A node is colored orange when it has been selected for visualization in the subnetwork view. A yellow node represents the current ego selected for the radar chart visualization.

network has the small-world property, it is possible to cover a large portion of nodes and edges with a comparatively small subset of ego-graphs. That is, the problem can be translated into the Set Cover problem. Since this is an NP-hard problem [238], we use a heuristic approach. We calculate the ego-graphs for every node in the network and sort them by their cardinality. Then, we take the largest ego-graph and remove the covered edges from the remaining ego-graphs. We repeat this step until either a specified threshold of interaction coverage (default: 90 %) or a predefined maximum number of ego-graphs (default: 100) is reached.

The resulting overview network of relevant ego-graphs is visualized using the aggregated ego-graph glyphs (Figure 7.4). The percentage of nodes and edges in the original network covered by the resulting overview ego-graph network is displayed as a text label at the top. Following the "show context" and "details on demand" principles, each node in the overview can be selected for further inspection in the other views (**R2**). Moreover, the coloring of the aggregated glyphs in the overview provides context for the current selection for the visualizations. Glyphs are colored orange if the corresponding ego is visualized in the ego-graph subnetwork, and yellow if it is visualized in the radar chart. Ego-graph glyphs in the overview that are not selected are colored using a white-to-gray gradient, illustrating the percentage of nodes in the ego-graph that are contained in the ego-graph subnetwork (*0%* *100%*). This allows users to either focus on ego-graphs that have a high overlap with the current selection (dark gray) or highly dissimilar ones (white or light gray), depending on their current analysis task.

Radar Chart

The *radar chart* provides information about a metadata attribute of egos whose ego-networks are similar to the one of the selected ego. (Figure 7.1b, **R4**). Similar to the aggregated glyphs, each circle represents an ego-graph, with the area corresponding to its cardinality. The radial distance to the center encodes the Jaccard index between the ego-graphs, i.e., the closer a node is to the center, the more alters it shares with the selected ego. This places the radar chart in close relation to the concept of *monadic exploration* [239]. The

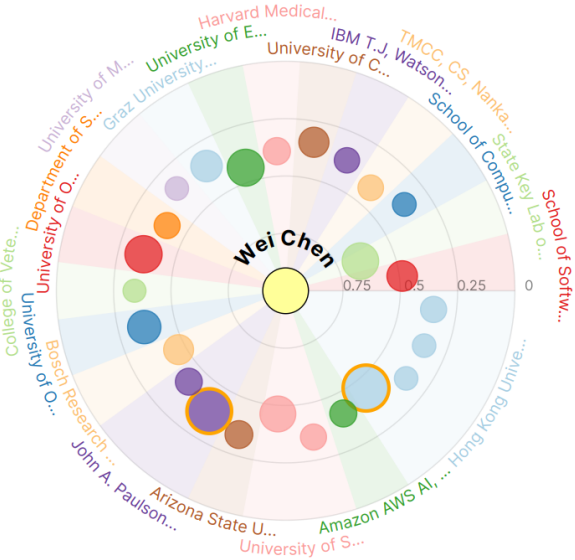


Figure 7.5: Similar ego-graphs to a reference ego-graph (radar center) in a co-authorship network classified by affiliation. The radar chart shows the 25 ego-graphs most similar to the ego-graph in the center. The distance to the center corresponds to the Jaccard index. In addition to the similarity, categorical metadata is visualized. In this example, each circle represents the ego-graph of an author, while the colors represent their affiliation. Circles with an orange outline correspond to ego-graphs selected in the ego-graph subnetwork.

core *monadic exploration* is to take the viewpoint of a subnetwork and display other subnetworks with overlapping relevance radially around it. Topics of higher relevance are placed closer to the center than topics of lower relevance. To avoid clutter, we only show the n ego-graphs with the highest Jaccard index (default $n = 25$). The colors of the nodes represent the metadata associated with the egos, such as author affiliation in a co-author network or the BRITE functional hierarchy in the case of proteins [64]. Ego-graphs that belong to the same category are put next to each other, and the corresponding circular segment of the radar chart is colored semi-transparently with the same color. Additionally, text labels naming the categories corresponding to the circular segments are put around the radar chart. Users can select ego-graphs in the radar chart to add them to the ego-graph subnetwork. Ego-graphs in the radar chart that are also shown in the ego-graph subnetwork view have an orange outline, as shown in Figure 7.5.

Ego-graph Subnetwork

Ego-graphs selected in the overview or the radar chart are visualized in the *ego-graph subnetwork* (Figure 7.1c), showing different levels of details of the respective ego-graphs. As mentioned in subsection 7.4.1, the ego-graphs are initially visualized using the aggregated glyphs, but can be de-aggregated to the detailed glyphs on demand (Figure 7.6, **R3**). The color of each aggregated ego-graph glyph in this view encodes a quantitative metadata value associated with the ego (*min value*  *max value*). Up to three connected ego-graphs can form an ego-graph group, as explained in subsection 7.4.1.

Selection Table

Groups of nodes in ego-graphs or intersections can be selected for investigation in the selection table (Figure 7.7), shown on demand using a menu button. The table contains additional attributes for each node, such as metadata (**R4**) and information on the nodes, e.g., whether they are present in the overview and the ego-graph subnetwork. The rows can be sorted by any of the columns containing the attributes. The user can select any node for visualization in

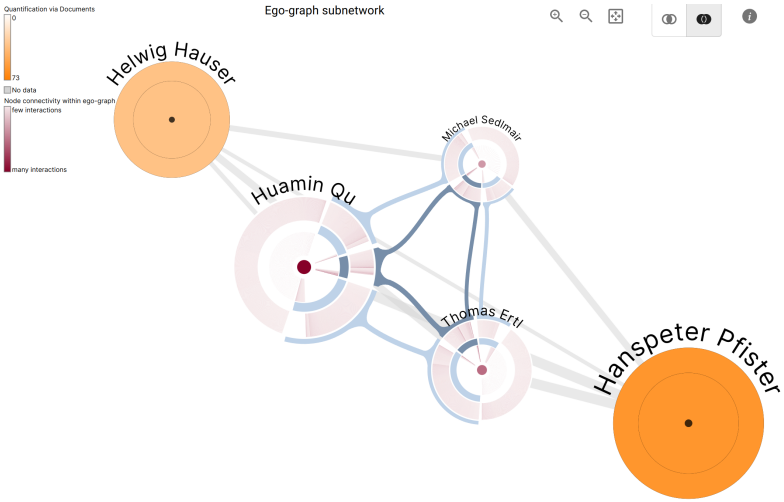


Figure 7.6: Ego-graph subnetwork of a co-author network with five ego-graphs. A group of three ego-graphs is shown in its detailed view. The width of the gray *similarity edges* encodes the similarity between ego-graphs outside ego-graph groups, while the blue *identity bands* link identical nodes within an ego-graph group.

COLUMNS 1 FILTERS

Radar	nodeID	institution	Documents ↓	Citations	Found in Ov…
☐	Kwan-Liu Ma	University of Califo…	73	1931	Yes
☐	M. Eduard Gröller	TU Wien, Austria	69	1680	No
☑	Huamin Qu	Hong Kong Univer…	68	2615	Yes
☐	Arie E. Kaufman	Department of Co…	61	788	Yes
☑	Hanspeter Pfister	John A. Paulson S…	60	3568	Yes
☐	Daniel A. Keim	University of Konst…	58	2304	Yes

Figure 7.7: Excerpt of the selection table showing the top 5 entries of the co-author data set (sorted by *Documents*, i.e., number of published papers). Only a subset of the columns is shown. The checkbox to the left adds the entry to the ego-graph subnetwork view.

the ego-graph network and the radar chart. For a detailed analysis of intersections between ego-graphs, the user can select the corresponding intersection band in the ego-graph subnetwork, which allows to filter and sort the table by this subset.

7.4.4 Implementation

We implemented `ProtEGOnist` as a web-based application with a client-server architecture. The server backend was written in Python using Flask [143]. The user interface and the visualizations in the frontend were mainly implemented in TypeScript using *React* [144], *Jotai* [240], *Material-UI* [241], and *D3* [146]. In the backend, we use the Python library *networkX* [242] for the extraction of relevant features from the graph structure. `ProtEGOnist` is available at `https://protegonist-tuevis.cs.uni-tuebingen.de/`.

7.5 Use Cases

In this section, we demonstrate the applicability of our approach using three use cases. The first one shows the utility of `ProtEGOnist` and the interaction of its components for exploring a co-author network. The other two use cases show how it can be applied to PPI networks. The PPI network of *E. coli* serves as a well-known example data set for domain experts from biology and highlights the advantages of the glyph design. The second PPI network stems from the Bio+MedVis Challenge 2023 and illustrates the application of `ProtEGOnist` to metadata-enriched data sets.

7.5.1 Co-author network

To showcase the usefulness of our `ProtEGOnist` approach for exploring social networks, we applied it to the Visualization Publications data set [206]. This data set contains all publications of the IEEE VIS conference (SciVis, InfoVis, VAST) and its predecessor symposia and conferences. The metadata for each entry includes, e.g., the authors and the number of publications. The resulting co-author network has 6,610 nodes and 22,220 edges. The network and the

metadata were extracted directly from the data, and the citation count provided by CrossRef [243] was used.

A typical starting question when exploring a co-author network could be to find out who the most well-connected researchers are, and whether they are also the most prolific ones in terms of publications. Investigating the ego-graph network using the *Network Overview*, the user can determine that the nodes for Huamin Qu, Hanspeter Pfister, and Wei Chen are the largest, indicating that they have the largest number of 1st and 2nd-degree coauthors (Figure 7.4, **R1**). We selected these three nodes for the *ego-graph subnetwork* view, which helps to visually compare node sizes (**R2**). The color mapping (*0* *max*) in the *ego-graph subnetwork* reveals that they all have a high number of publications (**R4**). Sorting the *Selection Table* by the number of documents (i.e., co-authored publications) allows for a quantitative assessment of the number of publications: all three are high-ranking, with Qu and Pfister being #3 and #5, respectively. Interestingly, Chen is only #13 (Figure 7.7), despite having a high number of co-authors. Adding the two top-ranking researchers concerning their number of publications—Kwan-Liu Ma and M. Eduard Gröller—for an in-depth comparison reveals that Chen has a larger network than Gröller but also a higher percentage of unique co-authors that are not shared by the two (Figure 7.8, **R3**). One reason for the comparably large co-author ego-graph of Chen might be his joint publications with Qu and Pfister, thus benefitting from their large networks. Exploring the radar chart reveals that Chen has also published with other well-connected researchers like David Ebert, Benjamin Bach, or Yingcai Wu (identified by hovering the largest nodes in the radar chart shown in Figure 7.5). It also shows that the ego-graph of Chen contains researchers from institutions from all over the world.

7.5.2 lac operon in E. coli Protein-Protein Interaction Network

For protein-protein interactions, specific proteins and their context are often of interest. In the bacterium *E. coli*, the lactose operon (short *lac* operon) is a well-studied set of proteins that is required

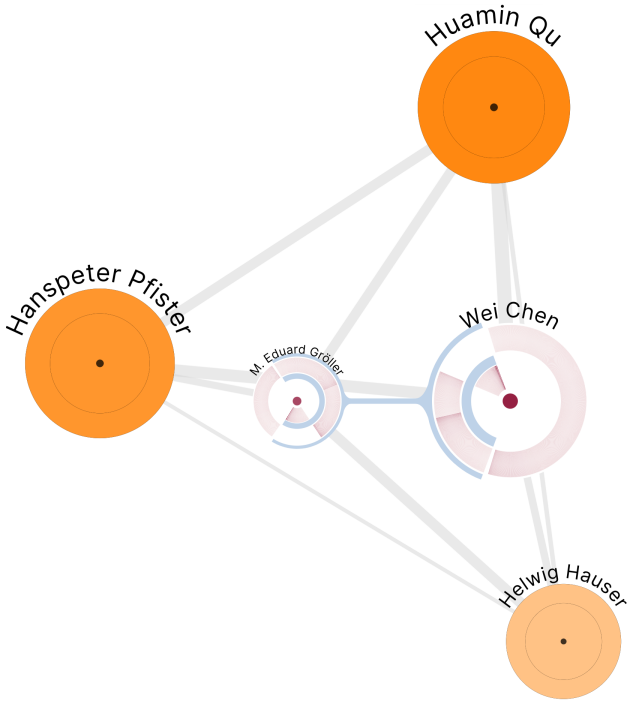


Figure 7.8: Ego-graph subnetwork visualization with a detailed ego-graph group of co-author networks of *Gröller* and *Chen*. The detailed glyphs reveal that Chen has not only a larger network in total but also a higher percentage of unique second-level co-authors.

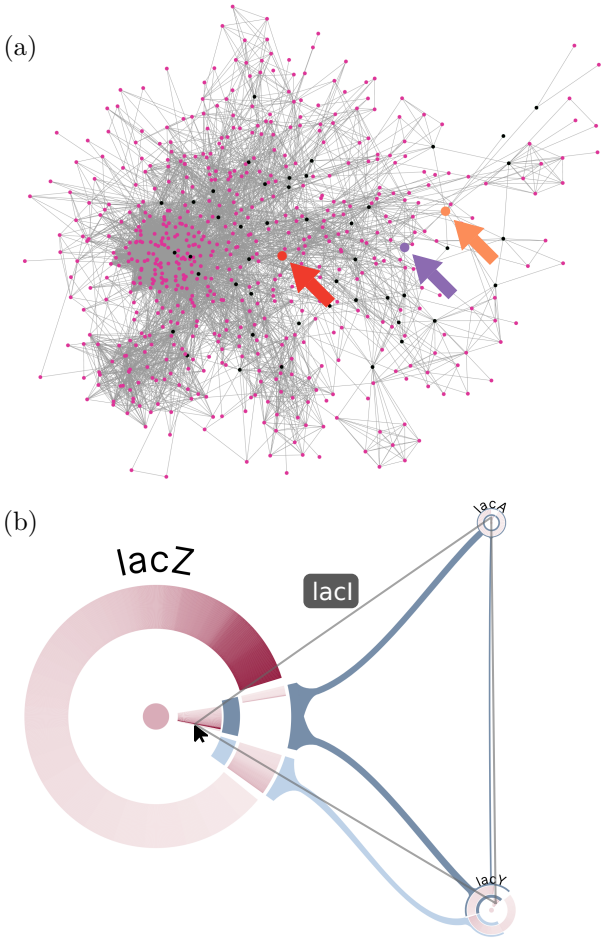


Figure 7.9: Visualizing the *lac* operon of *E. coli* using a node-link diagram created with *Cytoscape* (a). *lacZ*, *lacA*, and *lacY* are colored in ●, ● and ●, respectively. Nodes of distance one are colored ● while the ones of distance two are colored ●. The network consists of 653 nodes and 8,435 edges. Only interactions with a confidence score higher than 0.75 were considered. Visualizing the same proteins in `ProtEGOnist` (b). The node corresponding to *lacI* is hovered and shown in all ego-graphs.

for the metabolism of lactose. It is active if glucose, the preferred energy source, is not available but only lactose.

Here, we analyze the PPI network of the K12 strain of *E. coli* as found in the *STRING* database [22] and demonstrate the effect of the ego-graph layout for analyzing three proteins in detail (**R3**). As a baseline, we loaded the PPI network into *Cytoscape* [208]. Figure 7.9a shows the proteins lacZ, lacY, and lacA of the *lac* operon and their degree-1 and degree-2 alters in a simple node-link diagram created using the *Cytoscape StringApp* [244]. The node-link diagram forms a hairball-like structure due to the high number of nodes and edges. We can see that there is only a comparatively low number of degree-1 alters to the three proteins of interest (black nodes in Figure 7.9). Moreover, an edge connecting lacA and lacY—indicating a direct interaction between the two proteins—is visible. Any other conclusions about the connectivity between the lac proteins or about the sizes of the individual neighborhoods cannot be made due to occluding edges.

In comparison, with `ProtEGOnist` the neighborhood of the lac operon proteins can be grouped into three ego-graphs (Figure 7.9b). Strikingly, we can see that lacZ has by far the largest ego-graph of which most degree-2 alters are unique. This shows that lacZ also interacts with proteins not directly involved in the *lac* operon, indicating that it has a more central role in the PPI network compared to the other two proteins lacA and lacY. In contrast to lacZ, lacA has no unique alters, indicating a role more restricted to the operon.

From the band coloring, we can conclude that a large proportion of proteins is shared between all three ego-graphs. Notably, most of them have a distance of one to lacZ, while they have a distance of two to the other proteins. In fact, by hovering over the proteins, we find that the only degree-1 alter shared by all three proteins is lacI, which serves as the repressor for the operon.

Using the degree-2 alters, more distant associations can be investigated, for example, the relationship of the lac operon and the citrate cycle. The citrate cycle is one of the central metabolic pathways providing energy to the cell. When comparing the ego graphs

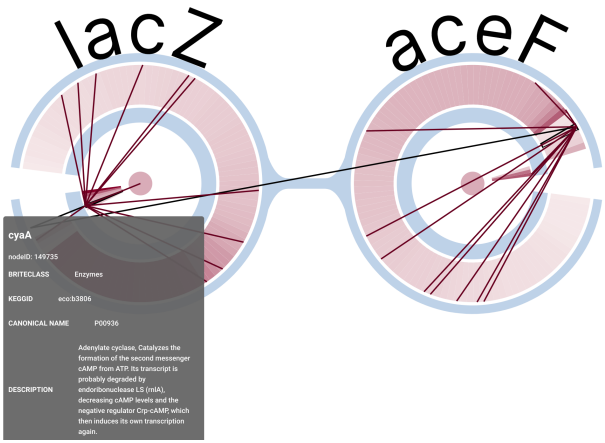


Figure 7.10: The ego-graph subnetwork of nodes shared by *lacZ* and *aceF*. When hovering over any degree-1 alters of the two egos, the identity edges are shown. This interaction is shown for *cyaA*. With this interaction, it can be seen that none of the degree-1 alters are found within the degree-1 alters of the other ego graph.

of *lacZ* and *aceF*, a pivotal enzyme in the citrate cycle, by investigating the respective degree-1 alters we can see that they only have multi-degree associations (Figure 7.10).

7.5.3 Human DeeProM Protein-Protein Interaction Network

We used the current version of `ProtEGOnist` to analyze the data set by Gonçalves *et al.* [204] originally provided for the Bio+MedVis Challenge 2023. Proteins are common drug targets, i.e., drugs modify proteins to cause changes in the cell. In the case of cancer, drugs aim at disturbing the molecular pathways in cancer cells while leaving non-cancerous cells widely unharmed. Gonçalves *et al.* used a deep-learning approach to identify associations between drugs and proteins.

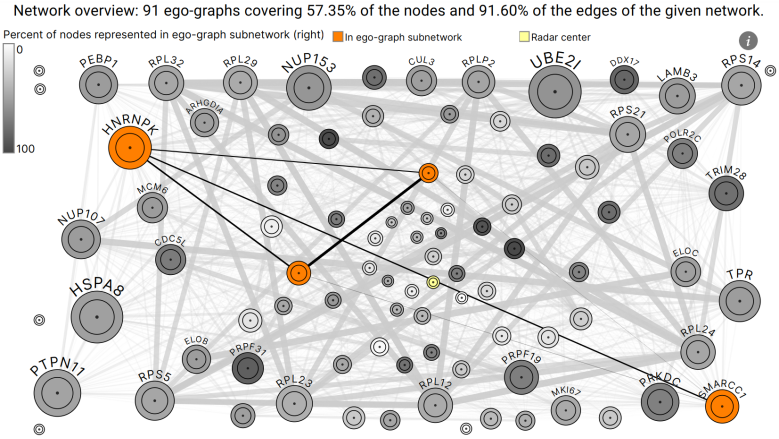


Figure 7.11: Network overview for the DeeProM data set. The set of 91 ego-graphs is based on the proteins identified by Gonçalves *et al* [GPC*22] as part of the 108 most-prominent protein-drug associations of their data set. These cover 57.35% of the nodes and 91.60% of the edges of the original network. The color scale from white to gray corresponds to the percentage of the nodes in the ego-graphs currently visualized in the ego-graph subnetwork. A node is colored orange when its ego has been selected for visualization in the subnetwork. A yellow node represents the current ego selected for the radar chart visualization.

Analysis Using ProtEGOnist

For the overview, the ego-graphs of 91 proteins identified in the original publication to have relevant drug-protein associations were chosen (Figure 7.11). This is an alternative approach to the other use cases, where the ego-graphs were selected via our set cover heuristic. Our analysis revealed that the union of proteins contained in these ego-graphs covers 57.3% of the proteins (nodes) and 91.6% of the interactions (edges) in the original PPI network. That is, the ego-graphs based on the proteins identified by DeeProM reflect most of the interactions in the original PPI network (**R1**). The metadata loaded into ProtEGOnist containted the drug-protein associations and the BRITE classification of the proteins.

Using ProtEGOnist, the results of DeeProM can be explored, opening up the black-box deep-learning model. Users can explore the proteins in the overview network in more detail, e.g., by selecting those associated with one drug of interest and viewing their BRITE functional classification. For the drug *Ara-G*, which prevents the elongation of DNA of cancer cells, four associated proteins are found in the overview network. Further inspection of these proteins in the subnetwork reveals three highly connected ego-graphs and a more distant one. The three most highly connected ego-graphs were selected as an ego-graph group (Figure 7.12). The lesser connected protein *SMARCC1* has been identified as a suppressor in some types of cancers [245], while the others act as possible drug targets or cancer biomarkers [246]–[248]. All three proteins are associated with the BRITE class *Spliceosome*, i.e., these proteins are involved in the maturation of mRNA before translation [249]. This association is even more prominent when inspecting the proteins *PPIH* and *RBM39* using the radar chart (Figure 7.13, **R4**). In the ego-graph group, similarities of the highly connected ego-graphs can be explored in more detail by viewing shared proteins when selecting the intersection of all three ego-graphs (**R3**). From this point, users could, e.g., continue analyzing the found proteins using the KEGG pathway annotations to see in detail how they relate functionally.

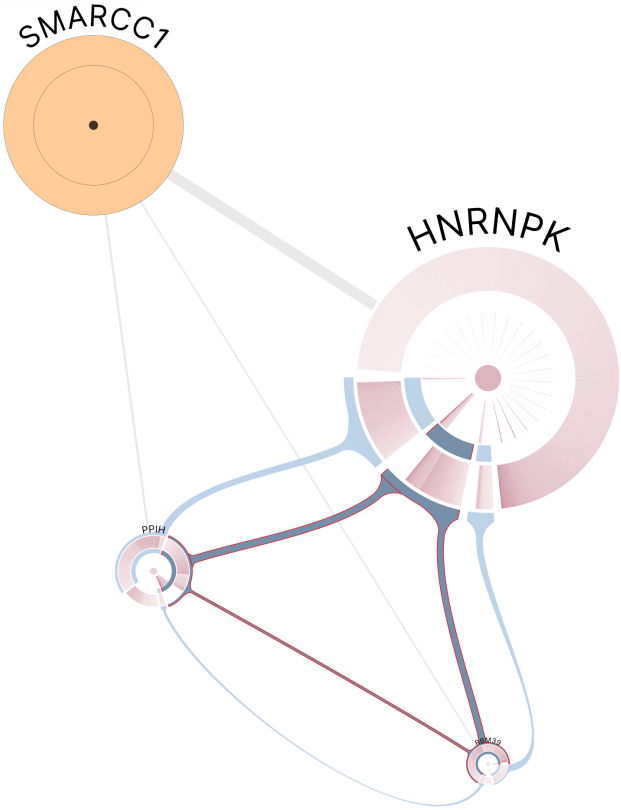
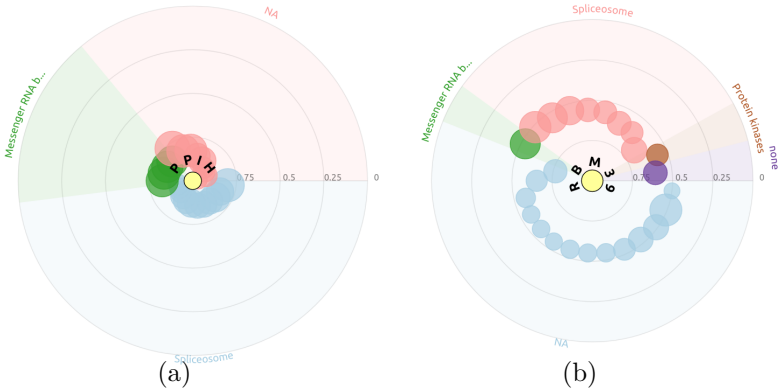


Figure 7.12: Ego-graph subnetwork for all proteins of the overview network associated with the drug *ARA-G*. The three most highly connected ego-graphs (*PPIH*, *HNRNPK*, and *RBM39*) are shown as an ego-graph group. The intersection between all three ego-graphs has been selected (red outline).



(a) (b)

Figure 7.13: Radar charts for the proteins (a) *PPIH* and (b) *RBM39*. The closer the circles are to the center of the radar chart, the more similar the ego-graphs.

Expert Feedback

Due to the positive outcome of the Bio+MedVis Challenge 2023 [203], we contacted the authors of the DeeProM data set to get their expert feedback on `ProtEGOnist`. We demonstrated our application to three of them and got a very positive response. One of the authors volunteered to test our application and to provide feedback. It consisted first of the free exploration and visualization of the data set using `ProtEGOnist`. Subsequently, we provided a structured questionnaire of ten questions based on the System Usability Scale (SUS) framework [151] and further open-ended questions. The expert that evaluated our tool had explored similar data sets using visualizations provided by *STRING* [22] and *Reactome* [250]. `ProtEGOnist` was assessed as slightly cumbersome, but they considered the complexity of our approach necessary, and its learning curve not steep. Overall, the expert enjoyed the exploration using ego-graphs. Nevertheless, they missed the integration of further information, e.g., which pathways a protein is involved in. As `ProtEGOnist` is easily extendable with additional arbitrary metadata, the pathway annotations were added as a new column to the selection table by adapting the input data.

7.6 Discussion

We presented `ProtEGOnist`, an interactive approach that applies ego-graphs to small-world interaction networks. Ego-graphs are a concept often encountered in real life, for example, when thinking about own friends or friends of friends in social networks. Therefore, the application of ego-graph in other domains, such as biological networks, employs a well-known mental model. In our case studies, we showed that this concept can be applied to data sets for a broad audience, like social networks, as well as to more domain-specific problems, such as PPI networks.

As shown in the co-author use case, the approach can be used to explore the network from an overview level down to detailed groups of ego-graphs. Furthermore, for the overview, we exploit the small-world property, which states that the maximal distance between two nodes is small compared to the network size. Therefore, it is possible to show a relatively small set of ego-graphs as an overview while covering a large portion of the original interactions. Although the co-author network consists of 6,610 nodes and 22,220 edges, 100 ego-graphs suffice to cover more than 90 % of interactions and almost 80 % of nodes. Conceptually, even larger networks could be displayed using `ProtEGOnist`, as long as it fulfills the small world property. However, for very large networks, the overview might not be able to cover a large proportion of the network as the average minimum distance between the nodes could be too large. While 2-level ego-graphs are commonly used in practice, in the case of larger networks, ego-graphs with further levels of alters might be more appropriate.

In contrast to the *overview-first* approach of the first use case, in the second use case, we started the analysis with previous knowledge and analyzed the *lac* operon of *E. coli* in the context of the network. Here, we demonstrated the scalability of the glyph layout. In contrast to the conventional node-link diagram, `ProtEGOnist` allows us to immediately assess the size of the ego-graphs, and thus the centrality of the protein in the network. Even though the *lacZ* ego-graph contains 620 nodes, the layout effectively groups the nodes into sections containing unique or shared nodes, and alter levels.

The sorting within the sections subsets the data even further and visualizes the distribution of the node degrees. We also illustrated the usefulness of 2-level ego-graphs for inspecting distant associations.

As exemplified in the third use case, our approach can easily be used with a user-defined set of nodes in the ego-graph overview network. Moreover, this example shows how network exploration in `ProtEGOnist` can be enhanced with different kinds of metadata. For PPI networks, even more data can be included in the analysis for visualization in the radar char or the aggregated ego-graphs of the subnetwork. For example, further omics data, like gene expression data or genomic data on mutations, could be used to analyze the proteins in more detail. This flexibility concerning the input data shows that `ProtEGOnist` can be generalized to a wide variety of application areas in which the small world property is fulfilled, e.g., linguistics [251], computer networking [252], or transportation networks [253].

Apart from `ProtEGOnist`, only few approaches have been proposed for comparing ego-graphs. However, their underlying goals are only remotely related to our approach. Out of those approaches, many are tailored to the visualization of dynamic networks. `EgoLines` [202], among others [227], [228], utilizes a linear layout of subsequent stages of the same ego-graph for a direct comparison of a stage with its predecessor or successor. `ProtEGOnist` aims at comparing different ego-graphs in a non-dynamic context. While the comparison in dynamic networks is often focused on the gradual changes of a single ego-graph, the differences when comparing multiple ego-graphs can be substantial. `EgoComp` [221] tackles this task for two ego-graphs. Our tool extends this approach to compare up to three ego-graphs simultaneously and puts them in context with other ego-graphs. We deliberately chose the comparison of three ego-graphs, as this allows us to use a layout of the setwise intersections without edge crossings. Increasing the amount of ego-graphs would be possible but incur edge crossings. Moreover, due to the usage of bands instead of a conventional node-link diagram, `ProtEGOnist` has a much clearer layout for large ego-graphs.

7.7 Outlook & Conclusion

In the future, we plan to extend `ProtEGOnist` with more ways to incorporate metadata into the analysis process. One direction would be to associate edges and bands with metadata and to include separate visualizations for metadata, as well as network filtering options based on the metadata.

Moreover, we plan to allow the upload of user-generated data. By providing a network structure, metadata, and, if available, a set of nodes of interest, our approach can then be used for many other small-world network cases, such as transportation networks. We also plan to generalize our approach to accept different distance metrics to substitute the Jaccard index as the distance value. One of the improvements we identified from the expert feedback was the lack of connection between the visualization components and the tabular view. To enhance this connection, we plan to include a *pop-up* interaction for the selection table, where it is shown automatically when selecting a specific ego-graph or intersections between ego-graphs in the ego-graph subnetwork.

With `ProtEGOnist`, we provide a layout focused on ego-graphs omitting edges and aggregating subnetworks into single glyphs. We believe that the novel layout is one of the main reasons that our approach was rated as being a bit cumbersome to use. Adding a conventional node-link diagram to visualize one or more ego-graphs (similar to `STRING` [30]) could support the exploration process by providing a less abstract visualization as a detail view. These node-link diagrams could be shown either for the currently selected subnetwork or for a single ego-graphs, similar to `BioLinker` [234]. Without specialized layout techniques, however, even a small number of ego-graphs can lead to unreadable visualizations due to a high number of nodes and edges (Figure 7.9 a).

To conclude, `ProtEGOnist` fills a gap in the network research space by combining established concepts for the analysis of small-world networks to a novel visualization approach. While it was initially intended for a specific domain with well-defined tasks based on the Bio+MedVis Challenge 2023, we show that this approach is applicable to different small-world networks from various domains.

Acknowledgements

We would like to thank the chairing committee behind the Bio+MedVis Challenge 2023 for motivating the challenge and supporting the communication with the authors of the original data set. We would like to thank Zhaoxiang Simon Cai for providing expert feedback on `ProtEGOnist`.

Chapter 8

Discussion

This dissertation presented visualization approaches, each illuminating different scales, layers, and properties of multi-omics data. Each approach targets the visualization of complex high-dimensional multi-omics data from a different perspective based on the underlying biological questions. This chapter summarizes the extent to which the presented work covers the field of multi-omics data visualization. Furthermore, it highlights future directions in the field.

When using the term "multi-omics visualization" I refer to methods that visualize more than one omics layer simultaneously (as described in chapter 5). Using this definition, Figure 8.1 summarizes the multi-omics data that can be visualized with each approach. All presented tools visualize three or more omics layers. Yet, the scales and types of data differ, as the tools require different levels of preprocessing (measured data) or visualize curated data from databases (knowledge-based data). Furthermore, the tools integrate multi-omics data differently using either knowledge-based integration, data-driven aggregation, or composite networks. In the following, for each approach the data, the type of integration, and visualization are discussed, and how they contribute to a holistic multi-omics analysis.

`SeMa-Trap` presented in chapter 3 processes raw RNA-seq data of a single organism at multiple conditions and visualizes the results. It is therefore easy to apply for users with little computational skills. The tool integrates genomic data as a reference genome (either from a database or self-provided), predicts biosynthetic gene clusters, and thus indirectly incorporates metabolites. The approach represents knowledge-based data integration as it uses antiSMASH for the prediction of BGC regions with profile hidden Markov models of genes known to be specific for secondary metabolites [63], [115]. The

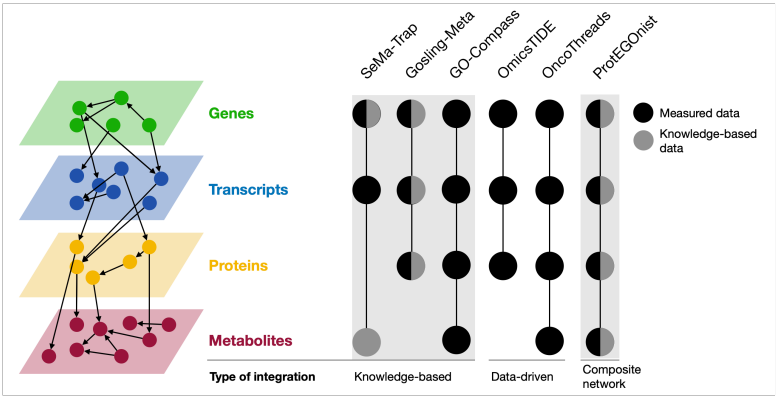


Figure 8.1: Multi-omics layers covered by the presented visualization approaches. The approaches all cover multiple omics layers. Some approaches explicitly require measured data for the omics layers, others visualize knowledge-based data from databases as well. Each approach integrates multi-omics data differently. The types of integration can be grouped into knowledge-based integration, data-driven integration, and composite networks.

genome is filtered to display an overview of the expression for each predicted BGC region to restrict the visualization to the most relevant features for the underlying biological question. From a visualization perspective, incorporating further omics layers would be relatively straightforward, but would also require much more computational processing. `Gosling-Meta`, in contrast, is universally applicable and does not have strong requirements concerning the data formats. Yet, using it requires computational skills, as it involves processing the data to a usable format and implementing the JSON specification. Therefore, it represents more of an approach for developers than for domain experts.

`GO-Compass` (chapter 4) is a knowledge-based integration approach that does not have a restriction for the supported omics layers. Any identifiers translatable into GO terms can be used. While this has been done mostly for gene identifiers, which makes this technique applicable to genomics, transcriptomics, and proteomics, recently overrepresentation tests have also been applied to metabolomic data [254]. The approach of `GO-Compass` applies both filtering and aggregation techniques for visualization. Aggregation is applied with GO terms which summarize the function of groups of genes. Furthermore, redundant GO terms are filtered out, so that the visualization is focused on the most relevant information.

In chapter 5 `OmicsTIDE` was presented, which visualizes experimental data on the scale of multiple omics layers. It applies a data-driven integration approach by jointly clustering multi-omics data with shared gene identifiers. Therefore, it is limited to abundance data directly mappable to genes, such as copy number variation in genomics and transcript or protein abundances in transcriptomics and proteomics. The tool applies aggregation on different levels to provide an overview. It visualizes how many genes follow the same or a different trend in two data sets and aggregates groups of genes sharing a trend into a single visualization.

In contrast, `OncoThreads` (chapter 6) maps data to patients and therefore does not have a restriction on the types of omics layers. However, patients add a further scale to the visualization. It is an exploratory data-driven approach that aggregates cohorts of patients into groups according to multi-omics data or metadata. As

the patient cohort is visualized at different stages and screen space is limited, only a relatively small number of features can be displayed simultaneously. Computationally aided visualization methods help filter and identify the most interesting features.

`ProtEGOnist`, presented in chapter 7 is the most universal approach in this dissertation. As a general network visualization tool, it can indirectly incorporate any omics layer, since composite multi-omics networks such as those provided by `STRING` can be displayed. The approach uses aggregation to visualize entire small-world networks but also applies filtering to filter out nodes with few interactions. In the publication, it has been used to visualize a protein-protein association network for every protein in a species. Furthermore, it can incorporate abundance values and map them on the network nodes. Yet, due to its generalizability, it can be applied to multi-omics data on different scales, such as single-omics or multi-omics correlation networks. In any case, it requires processing the data in the correct formats, which makes it less readily applicable than specialized approaches with integrated pipelines. However, as exemplified in the paper by visualizing the entire PPI network of *E. coli* K12, it can then be used to display large curated data from databases, which is of interest to many researchers.

The research focus of the presented approaches is at the interface of biology, bioinformatics, visualization design, and software development. This is illustrated by the conferences and journals chosen for presentation and publication. Three of the presented approaches were published as tools in journals with a bioinformatics focus. `OncoThreads` was presented at ISMB 2021 and published in a special issue of Bioinformatics. `OmicsTIDE` and `SeMa-Trap` were published in the journals Bioinformatics Advances and Nucleic Acid Research, respectively. In contrast, the *State of the Art Report* on genomics, `GO-Compass`, and `ProtEGOnist` were presented at the visualization conference Eurovis (2019, 2023, 2024) and published in Computer Graphics Forum. While journals focused on the domain often require functioning tools with proof of concepts using real use cases, visualization venues are rather focused on novel visualization techniques and their evaluation. As the desired publication venue was

not always known from the beginning of the development of a technique, we often focused on both, the novelty of the visualization technique and its applicability to real data. However, prioritizing one or the other can enhance the development process, as was the case for `SeMa-Trap` and `ProtEGOnist`, both developed in a comparatively short time of less than a year. For `ProtEGOnist` we quickly identified its universality after the Bio+MedVis Challenge and thus shifted the focus of the application to networks in general using a well-known social-science data set. The aim for the approach `SeMa-Trap` was publishing it in a special issue of NAR from the beginning, as the main focus was providing a computational pipeline for biologists enhanced with visualization. This tool profited from the *State of the Art Report* as previous work in the visualization field, which created a taxonomy facilitating the development of novel approaches.

As this work combines multiple research fields, it can be evaluated from each perspective and each field provides opportunities for future work. While the individual chapters describe future work targeted to the individual approaches, here further general opportunities of the research fields are evaluated. From a domain perspective, the question of how to find a suitable tool for the data at hand needs to be discussed. Moreover, the general directions concerning visualization, multi-omics integration algorithms, and evaluation are discussed. Finally, I want to provide an outlook on how to guarantee the maintainability of approaches in the future.

The goal of our contributions to the transregional collaborative research center TRR 261 “Cellular Mechanisms of Antibiotic Action and Production” concerning multi-omics analyses was to provide a multi-omics overview of the analyzed data to make it interpretable for domain researchers. Yet, the collaborations have shown that a one-size-fits-all solution is hard to realize as the questions and available omics layers are diverse. This led to the development of a multitude of tools, each suited for different questions and data. While `SeMa-Trap` was developed specifically for the visualization of elements of interest (BGCs), `GO-Compass`, `OmicsTIDE` and `OncoThreads` were designed for exploration of entire experimental data

sets. `ProtEGOnist` and `Gosling-Meta` are generalizable approaches that can be used for a variety of input data.

Choosing the downstream analysis process and appropriate tools is a non-trivial task, requiring collaboration of researchers from different fields. Data analysis projects usually involve domain experts and computational researchers. Domain researchers plan and conduct experiments, computational researchers assist in analyzing the data produced by the high-throughput methods. Standardizing the requirement formulation, the data analysis process, and the presentation of results can streamline projects and facilitate defining downstream analysis goals. Shared vocabulary must be established for domain researchers and computational researchers as a first step. For this, computational researchers need to understand the biological questions, and domain researchers should have basic insights into the computational analysis process. For example, metadata often serves as documentation for domain researchers, while it should be machine-readable for computational researchers. Similarly, manual annotations of genes often do not fit the standards for computational processing. Together, researchers choose the downstream analysis approaches. Here, the task of the computational researcher is selecting and applying the analysis pipelines addressing the biological questions. The results must be presented in an easily accessible form, for example by hosting an automatically created and manually adapted analysis report for the conducted pipeline, such as the one provided by `nf-core` [255]. Ideally, this interactive report would link the conducted analyses to the appropriate exploratory visualization tools.

To properly understand visualizations and the produced results, the data processing has to be transparent and reproducible. Most of the presented visualization approaches include help texts or introductory texts for understanding the data processing and allow downloading the visualizations for communication of the findings made. However, further methods for data provenance tracking could be implemented. This relates to both, explaining the algorithms and pipelines involved in processing the data and the steps performed in exploring the data with the visualization.

While explaining the computational background in detail can enhance the interpretation, it often is impracticable as it adds to the complexity of the visualization. Therefore, many visualizations presented provide a simplified, abstract intuition for the algorithms. For example, `GO-Compass` clusters GO terms hierarchically by their dispersibility and relatedness, but details on how these factors contribute to the clustering are not provided in the visualization. Similarly, `ProtEGOnist` selects the most influential aggregated meta-nodes for an overview visualization, but it does not detail how exactly this selection is performed. While this might be sufficient for most users and most of the time, sometimes users might want to explore the provenance of a data point in more detail, especially when the algorithm produces a counterintuitive result. It can happen, for example, that a user does not intuitively agree with the dispensability of a node in `GO-Compass` or its position in the tree. Similarly, it can happen that `ProtEGOnist` does not rate a meta-node as influential and does not place it in the overview although a user expects this property. In explainable AI, visual analytics methods are used to visualize algorithmic outcomes [256], for example when analyzing a clustering result in detail [257]. Similarly, for `ProtEGOnist` and `GO-Compass` additional views could be implemented to show how the underlying algorithm processed the point of interest. For `GO-Compass` this would mean that the factors distinguishing a node from its closest relative are detailed. For `ProtEGOnist` the differences of a node of interest to the node rated as more influential in the overview would be detailed.

For the exploration process within the visualization, such as tracking which meta-nodes have been analyzed using `ProtEGOnist` or which clusters have been viewed in detail in `OmicsTIDE` provenance tracking could be applied, for example, using the library `Trrack` [187]. This means that each step in the exploration is saved and users can return to previous steps at any point and share the entire exploration process with others instead of only the findings. Naturally, this adds complexity to the exploration process and the implementation, as every step has to be stored, even exploration routes that do not lead to insight.

In general, the work presented is focused on the visualization of multi-omics data and applies different algorithms for data integration. Especially for data-driven integration as applied in OmicsTIDE and OncoThreads more advanced algorithms, such as machine learning approaches can help integrate multi-omics data in different ways. For example, the k-means clustering in OmicsTIDE is a clear candidate for replacement. However, machine learning algorithms are most commonly used when aiming to associate large multi-omics data consisting of a high number of samples with phenotypes, which makes OncoThreads a suitable candidate for extension. In fact, OncoThreads has already been extended with more advanced methods in the tool ThreadStates which applies hierarchical agglomerative clustering [12]. While machine learning methods can add to successful exploration, they increase complexity and thus the need for applying visualization methods for explainability and training for researchers in interpreting the results.

The visualization design was strongly influenced by collaborations with domain experts. OmicsTIDE, GO-Compass, and SeMa-Trap were all developed together with domain experts from the TRR and with data produced in the research projects. For some of the approaches, further structured input from domain experts in the development of the visualization could have sped up and enhanced the process. This is demonstrated for OncoThreads where we applied the Design Sprint technique as a structured approach for application design, which included interviewing domain experts to define the tasks and evaluating a prototype with users.

Most of the published prototypes or tools have been evaluated qualitatively. GO-Compass has been evaluated in semi-structured interviews with five researchers from biology, medicine, and bioinformatics. Similarly, for ProtEGOnist we received expert feedback. However, only quantitative user studies with statistical evaluation where approaches are compared can prove whether a method has a measurable impact. Yet, visualization approaches for complex data have several properties that make systematic evaluation hard. First, quantitative evaluation is time-intensive and the visualization might need to be revised due to the study's results. This is, of course, the purpose of a user study, but it is less attractive because qualitative

evaluation is often sufficient to obtain a publishable result. Second, visualization approaches are inherently hard to compare, as they are created based on the need for novelty. Therefore, it is usually hard to find a point of comparison. Third, even if a point of comparison is found, the selection of potential test users is hard. Extensive user studies require a relatively large number of users and only specialized researchers have the required data literacy. Furthermore, the test users should ideally not have worked with either of the visualizations to be compared before, which further restricts the pool of participants. However, basic visual techniques, such as the effectiveness of different channels, the design of basic charts, and the impact of visualization on decisions are often evaluated. Using evaluated techniques and explicitly highlighting their effects on users in publications helps create sound and credible methods.

The presented approaches are at least on a functional research prototype level and can be used by domain experts without installation as they are web-based. Many approaches are hosted on the TueVis web server, which was born as a place to host `OmicsTIDE` but quickly grew into a solution used by many bioinformatics groups in Tübingen. The web server also hosts a website classifying the approaches by omics layers and other properties. Therefore, users can quickly find solutions appropriate for their data. This functionality will be extended to show additional information on shared properties of the hosted software, such as details on input data. Furthermore, we aim to enable data transfers between the tools. This means that, for example, gene lists that are the output of `OmicsTIDE` can automatically be passed to `GO-Compass`.

An approach should not only be usable by end-users but should also be easily extendable for software developers and researchers. Thereby, a technique can be extended and improved even if it is no longer actively maintained. `Gosling` aims at being a universal technique and has been extended in different ways, including `Gosling-Meta` presented in this dissertation. However, this is still to be done with other approaches presented. For example, the visualizations of `GO-Compass` could be made available as a Python package, where they could be more easily integrated with existing

data analysis pipelines. Similarly, the `ProtEGOnist` network algorithm could be separated from the visualization and integrated into existing network tools, such as `Cytoscape` [208].

Any approach requires maintenance after publication. The technologies used for web development, such as the programming languages, libraries, browsers, and back-end technologies are continuously updated to ensure safety and to implement novel, useful features. Updating the web applications using these techniques is crucial to keep them usable. Furthermore, users might discover bugs that need to be fixed. Yet, in scientific research, an approach often only survives for as long as the creator—often a single person—supports it. The focus when developing a technique is often generating a publishable result as quickly as possible. This often results in implementations that lack tests, are hard to understand for outsiders, and are not easily extendable. This is not necessarily problematic, since a throw-away prototype is often sufficient to evaluate a technique. Clean reimplementations of promising tools are often less time-consuming than anticipated and can greatly enhance the maintainability of an approach. However, even if an approach is well-implemented and if evidence exists that it is indeed useful, for long-term support, other structures and personnel are needed for the maintenance and ongoing development of tools. This means, that ideally, approaches should be developed and maintained by teams which have redundancies in their knowledge about the approach. Yet, this is not something that is easily supported by funding.

In conclusion, this dissertation illustrates how multi-omics data visualization can be tackled from different perspectives. Each presented technique has a unique focus and combines principles of visualization research with domain knowledge. While all approaches are implemented at least on a working prototype level, their long-term impact can be seen as reusable, well-demonstrated ideas. Yet, challenges remain on the interface of visualization research and biology, including making approaches well-known, explainable, and permanently available.

Bibliography

[1] S. Wilcken, P.-H. Koutsandrea, T. Bakker, *et al.*, "The TetR-like regulator Sco4385 and crp-like regulator Sco3571 modulate heterologous production of antibiotics in streptomyces coelicolor M512," en, *Appl. Environ. Microbiol.*, vol. 91, no. 5, e0231524, May 2025, ISSN: 0099-2240,1098-5336. DOI: 10.1128/aem.02315-24.

[2] L. Schulze, J. Moessner, S. Krauss, *et al.*, "Genetic modification of intractable staphylococcal clones by heat-shock facilitated phage transduction," *bioRxiv*, Apr. 2025. DOI: 10.1101/2025.04.04.647181.

[3] N. Gericke, D. Beqaj, T. Kronenberger, *et al.*, "Unveiling the substrate specificity of the ABC transporter tba and its role in glycopeptide biosynthesis," en, *iScience*, vol. 28, no. 4, p. 112 135, Apr. 2025, ISSN: 2589-0042. DOI: 10.1016/j.isci.2025.112135.

[4] S. Hackl, C. Jachmann, M. Witte Paz, T. A. Harbig, L. Martens, and K. Nieselt, "PTMVision: An interactive visualization webserver for post-translational modifications of proteins," en, *J. Proteome Res.*, Jan. 2025, ISSN: 1535-3893,1535-3907. DOI: 10.1021/acs.jproteome.4c00679.

[5] A. Hoffmann, U. Steffens, B. Maček, *et al.*, "The unusual mode of action of the polyketide glycoside antibiotic cervimycin C," en, *mSphere*, vol. 9, no. 5, e0076423, May 2024, ISSN: 2379-5042. DOI: 10.1128/msphere.00764-23.

[6] M. Bianchi, M. Winterhalter, T. A. Harbig, *et al.*, "Fosfomycin uptake in escherichia coli is mediated by the outer-membrane porins OmpF, OmpC, and LamB," en, *ACS Infect Dis*, vol. 10, no. 1, pp. 127–137, Jan. 2024, ISSN: 2373-8227. DOI: 10.1021/acsinfecdis.3c00367.

[7] R. Shukla, A. J. Peoples, K. C. Ludwig, *et al.*, "An antibiotic from an uncultured bacterium binds to an immutable target," en, *Cell*, vol. 186, no. 19, 4059–4073.e27, Sep. 2023,

ISSN: 0092-8674,1097-4172. DOI: 10.1016/j.cell.2023.07.038.

[8] M. D. Mungan, T. A. Harbig, N. H. Perez, *et al.*, "Secondary metabolite transcriptomic pipeline (SeMa-trap), an expression-based exploration tool for increased secondary metabolite production in bacteria," en, *Nucleic Acids Res.*, vol. 50, no. W1, W682–9, May 2022, ISSN: 0305-1048,1362-4962. DOI: 10.1093/nar/gkac371.

[9] S. Krauss, T. A. Harbig, J. Rapp, *et al.*, "Horizontal transfer of bacteriocin biosynthesis genes requires metabolic adaptation to improve compound production and cellular fitness," en, *Microbiol Spectr*, e0317622, Dec. 2022, ISSN: 2165-0497. DOI: 10.1128/spectrum.03176-22.

[10] S. T. Hackl, T. A. Harbig, and K. Nieselt, "Technical report on best practices for hybrid and long read de novo assembly of bacterial genomes utilizing illumina and oxford nanopore technologies reads," en, *bioRxiv*, p. 2022.10.25.513682, Oct. 2022. DOI: 10.1101/2022.10.25.513682.

[11] A. Dietrich, U. Steffens, M. Gajdiss, *et al.*, "Cervimycin-resistant staphylococcus aureus strains display vancomycin-intermediate resistant phenotypes," en, *Microbiol. Spectr.*, vol. 10, no. 5, e0256722, Oct. 2022, ISSN: 2165-0497. DOI: 10.1128/spectrum.02567-22.

[12] Q. Wang, T. Mazor, T. A. Harbig, E. Cerami, and N. Gehlenborg, "ThreadStates: State-based visual analysis of disease progression," en, *IEEE Trans. Vis. Comput. Graph.*, vol. 28, no. 1, pp. 238–247, Jan. 2022, ISSN: 1077-2626,1941-0506. DOI: 10.1109/TVCG.2021.3114840.

[13] M. Witte Paz, T. A. Harbig, and K. Nieselt, "Evidente—a visual analytics tool for data enrichment in SNP-based phylogenetic trees," en, *Bioinform. Adv.*, vol. 2, no. 1, Jan. 2022, ISSN: 2635-0041. DOI: 10.1093/bioadv/vbac075.

[14] S. Beier, A. Gorska, P. Grupp, T. A. Harbig, I. Flade, and D. H. Huson, "Bioinformatics support for the tubiom community gut microbiome project," en, PeerJ Preprints, Tech. Rep. e2382v1, Aug. 2016. DOI: 10.7287/peerj.preprints.2382v1.

[15] F. J. Anscombe, "Graphs in statistical analysis," *Am. Stat.*, vol. 27, no. 1, pp. 17–21, Feb. 1973, ISSN: 0003-1305. DOI: 10.1080/00031305.1973.10478966.

[16] J. R. Beniger and J. W. Tukey, "Exploratory data analysis," *Contemp. Sociol.*, vol. 7, no. 1, p. 64, Jan. 1978, ISSN: 0094-3061,1939-8638. DOI: 10.2307/2065930.

[17] M. Love, S. Anders, and W. Huber, "Differential analysis of count data – the DESeq2 package," *Genome Biol.*, 2013, ISSN: 1465-6906.

[18] H. Thorvaldsdóttir, J. T. Robinson, and J. P. Mesirov, "Integrative genomics viewer (IGV): High-performance genomics data visualization and exploration," en, *Brief. Bioinform.*, vol. 14, no. 2, pp. 178–192, Mar. 2013, ISSN: 1467-5463,1477-4054. DOI: 10.1093/bib/bbs017.

[19] A. Hartmann and A. M. Jozefowicz, "VANTED: A tool for integrative visualization and analysis of -omics data," en, *Methods Mol. Biol.*, vol. 1696, pp. 261–278, 2018, ISSN: 1064-3745,1940-6029. DOI: 10.1007/978-1-4939-7411-5_18.

[20] W. Luo and C. Brouwer, "Pathview: An R/bioconductor package for pathway-based data integration and visualization," en, *Bioinformatics*, vol. 29, no. 14, pp. 1830–1831, Jul. 2013, ISSN: 1367-4803,1367-4811. DOI: 10.1093/bioinformatics/btt285.

[21] T.-C. Kuo, T.-F. Tian, and Y. J. Tseng, "3Omics: A web-based systems biology tool for analysis, integration and visualization of human transcriptomic, proteomic and metabolomic data," en, *BMC Syst. Biol.*, vol. 7, p. 64, Jul. 2013, ISSN: 1752-0509. DOI: 10.1186/1752-0509-7-64.

[22] D. Szklarczyk, A. Franceschini, S. Wyder, *et al.*, "STRING v10: Protein-protein interaction networks, integrated over the tree of life," en, *Nucleic Acids Res.*, vol. 43, no. Database issue, pp. D447–52, Jan. 2015, ISSN: 0305-1048,1362-4962. DOI: 10.1093/nar/gku1003.

[23] T. Liu, P. Salguero, M. Petek, *et al.*, "PaintOmics 4: New tools for the integrative analysis of multi-omics datasets supported by multiple pathway databases," en, *Nucleic Acids Res.*, vol. 50, no. W1, W551–W559, Jul. 2022, ISSN: 0305-1048,1362-4962. DOI: 10.1093/nar/gkac352.

[24] S. Nusrat, T. Harbig, and N. Gehlenborg, "Tasks, techniques, and tools for genomic data visualization," en, *Comput. Graph. Forum*, vol. 38, no. 3, pp. 781–805, Jun. 2019, ISSN: 0167-7055. DOI: 10.1111/cgf.13727.

[25] S. L'Yi, Q. Wang, F. Lekschas, and N. Gehlenborg, "Gosling: A grammar-based toolkit for scalable and interactive genomics data visualization," en, *IEEE Trans. Vis. Comput. Graph.*, vol. 28, no. 1, pp. 140–150, Jan. 2022, ISSN: 1077-2626,1941-0506. DOI: 10.1109/TVCG.2021.3114876.

[26] T. Harbig, M. W. Paz, and K. Nieselt, "GO-Compass: Visual navigation of multiple lists of GO terms," en, *Comput. Graph. Forum*, vol. 42, no. 3, pp. 271–281, Jun. 2023, ISSN: 0167-7055,1467-8659. DOI: 10.1111/cgf.14829.

[27] T. A. Harbig, J. Fratte, M. Krone, and K. Nieselt, "OmicsTIDE: Interactive exploration of trends in multi-omics data," en, *Bioinform Adv*, vol. 3, no. 1, vbac093, Jan. 2023, ISSN: 2635-0041. DOI: 10.1093/bioadv/vbac093.

[28] T. A. Harbig, S. Nusrat, T. Mazor, *et al.*, "OncoThreads: Visualization of large-scale longitudinal cancer molecular data," en, *Bioinformatics*, vol. 37, no. Suppl_1, pp. i59–i66, Jul. 2021, ISSN: 1367-4803,1367-4811. DOI: 10.1093/bioinformatics/btab289.

[29] N. Brich, T. A. Harbig, M. W. Paz, K. Nieselt, and M. Krone, "ProtEGOnist: Visual analysis of interactions in small world networks using ego-graphs," en, *Comput. Graph. Forum*, vol. 43, no. 3, Jun. 2024, ISSN: 0167-7055,1467-8659. DOI: 10.1111/cgf.15078.

[30] C. von Mering, M. Huynen, D. Jaeggi, S. Schmidt, P. Bork, and B. Snel, "STRING: A database of predicted functional associations between proteins," en, *Nucleic Acids Res.*, vol. 31, no. 1, pp. 258–261, Jan. 2003, ISSN: 0305-1048,1362-4962. DOI: 10.1093/nar/gkg034.

[31] J. D. Watson and F. H. Crick, "Molecular structure of nucleic acids; a structure for deoxyribose nucleic acid," en, *Nature*, vol. 171, no. 4356, pp. 737–738, Apr. 1953, ISSN: 0028-0836. DOI: 10.1038/171737a0.

[32] F. H. Crick, "On protein synthesis," en, *Symp. Soc. Exp. Biol.*, vol. 12, pp. 138–163, 1958, ISSN: 0081-1386.

[33] E. S. Lander, L. M. Linton, B. Birren, *et al.*, "Initial sequencing and analysis of the human genome," en, *Nature*, vol. 409, no. 6822, pp. 860–921, Feb. 2001, ISSN: 0028-0836. DOI: 10.1038/35057062.

[34] F. H. Crick, L. Barnett, S. Brenner, and R. J. Watts-Tobin, "General nature of the genetic code for proteins," en, *Nature*, vol. 192, pp. 1227–1232, Dec. 1961, ISSN: 0028-0836. DOI: 10.1038/1921227a0.

[35] M. Kuroda, T. Ohta, I. Uchiyama, *et al.*, "Whole genome sequencing of meticillin-resistant staphylococcus aureus," en, *Lancet*, vol. 357, no. 9264, pp. 1225–1240, Apr. 2001, ISSN: 0140-6736. DOI: 10.1016/s0140-6736(00)04403-2.

[36] O. Rozenblatt-Rosen, A. Regev, P. Oberdoerffer, *et al.*, "The human tumor atlas network: Charting tumor transitions across space and time at single-cell resolution," en, *Cell*, vol. 181, no. 2, pp. 236–249, Apr. 2020, ISSN: 0092-8674,1097-4172. DOI: 10.1016/j.cell.2020.03.053.

[37] International Cancer Genome Consortium, T. J. Hudson, W. Anderson, *et al.*, "International network of cancer genome projects," en, *Nature*, vol. 464, no. 7291, pp. 993–998, Apr. 2010, ISSN: 0028-0836,1476-4687. DOI: 10.1038/nature08987.

[38] K. Tomczak, P. Czerwińska, and M. Wiznerowicz, "The cancer genome atlas (TCGA): An immeasurable source of knowledge," *Contemp. Oncol.*, vol. 19, A68–A77, Jan. 2015, ISSN: 1061-0383. DOI: 10.5114/wo.2014.47136.

[39] F. Sanger, S. Nicklen, and A. R. Coulson, "DNA sequencing with chain-terminating inhibitors," en, *Proc. Natl. Acad. Sci. U. S. A.*, vol. 74, no. 12, pp. 5463–5467, Dec. 1977, ISSN: 0027-8424. DOI: 10.1073/pnas.74.12.5463.

[40] D. R. Bentley, S. Balasubramanian, H. P. Swerdlow, *et al.*, "Accurate whole human genome sequencing using reversible terminator chemistry," en, *Nature*, vol. 456, no. 7218, pp. 53–59, Nov. 2008, ISSN: 0028-0836,1476-4687. DOI: 10.1038/nature07517.

[41] K. Mullis, F. Faloona, S. Scharf, R. Saiki, G. Horn, and H. Erlich, "Specific enzymatic amplification of DNA in vitro: The polymerase chain reaction," en, *Cold Spring Harb. Symp.*

Quant. Biol., vol. 51 Pt 1, pp. 263–273, 1986, ISSN: 0091-7451. DOI: 10.1101/sqb.1986.051.01.032.

[42] J. J. Kasianowicz, E. Brandin, D. Branton, and D. W. Deamer, "Characterization of individual polynucleotide molecules using a membrane channel," en, *Proc. Natl. Acad. Sci. U. S. A.*, vol. 93, no. 24, pp. 13 770–13 773, Nov. 1996, ISSN: 0027-8424. DOI: 10.1073/pnas.93.24.13770.

[43] W. Zhao, W. Zeng, B. Pang, *et al.*, "Oxford nanopore long-read sequencing enables the generation of complete bacterial and plasmid genomes without short-read sequencing," en, *Front. Microbiol.*, vol. 14, p. 1 179 966, May 2023, ISSN: 1664-302X. DOI: 10.3389/fmicb.2023.1179966.

[44] P. Flicek and E. Birney, "Sense from sequence reads: Methods for alignment and assembly," en, *Nat. Methods*, vol. 6, no. 11 Suppl, S6–S12, Nov. 2009, ISSN: 1548-7091,1548-7105. DOI: 10.1038/nmeth.1376.

[45] P. E. C. Compeau, P. A. Pevzner, and G. Tesler, "How to apply de bruijn graphs to genome assembly," en, *Nat. Biotechnol.*, vol. 29, no. 11, pp. 987–991, Nov. 2011, ISSN: 1087-0156,1546-1696. DOI: 10.1038/nbt.2023.

[46] H. Li and N. Homer, "A survey of sequence alignment algorithms for next-generation sequencing," en, *Brief. Bioinform.*, vol. 11, no. 5, pp. 473–483, Sep. 2010, ISSN: 1467-5463,1477-4054. DOI: 10.1093/bib/bbq015.

[47] D. A. Benson, M. Cavanaugh, K. Clark, *et al.*, "GenBank," en, *Nucleic Acids Res.*, vol. 41, no. Database issue, pp. D36–42, Jan. 2013, ISSN: 0305-1048,1362-4962. DOI: 10.1093/nar/gks1195.

[48] T. Hubbard, D. Barker, E. Birney, *et al.*, "The ensembl genome database project," en, *Nucleic Acids Res.*, vol. 30, no. 1, pp. 38–41, Jan. 2002, ISSN: 0305-1048,1362-4962.

[49] S. Andrews *et al.*, *FastQC: A quality control tool for high throughput sequence data*, https://www.bioinformatics.babraham.ac.uk/projects/fastqc/, Accessed: 2024-6-14, 2010.

[50] F. Krueger, "Trim galore!: A wrapper around cutadapt and FastQC to consistently apply adapter and quality trimming

to FastQ files, with extra functionality for RRBS data," *Babraham Institute*, 2015.

[51] M. Martin, "Cutadapt removes adapter sequences from high-throughput sequencing reads," *EMBnet. journal*, 2011.

[52] J. R. Miller, S. Koren, and G. Sutton, "Assembly algorithms for next-generation sequencing data," en, *Genomics*, vol. 95, no. 6, pp. 315–327, Jun. 2010, ISSN: 0888-7543,1089-8646. DOI: 10.1016/j.ygeno.2010.03.001.

[53] R. R. Wick, L. M. Judd, C. L. Gorrie, and K. E. Holt, "Unicycler: Resolving bacterial genome assemblies from short and long sequencing reads," en, *PLoS Comput. Biol.*, vol. 13, no. 6, e1005595, Jun. 2017, ISSN: 1553-734X,1553-7358. DOI: 10.1371/journal.pcbi.1005595.

[54] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool," en, *J. Mol. Biol.*, vol. 215, no. 3, pp. 403–410, Oct. 1990, ISSN: 0022-2836. DOI: 10.1016/S0022-2836(05)80360-2.

[55] A. L. Delcher, D. Harmon, S. Kasif, O. White, and S. L. Salzberg, "Improved microbial gene identification with GLIMMER," en, *Nucleic Acids Res.*, vol. 27, no. 23, pp. 4636–4641, Dec. 1999, ISSN: 0305-1048. DOI: 10.1093/nar/27.23.4636.

[56] C. Burge and S. Karlin, "Prediction of complete gene structures in human genomic DNA," en, *J. Mol. Biol.*, vol. 268, no. 1, pp. 78–94, Apr. 1997, ISSN: 0022-2836. DOI: 10.1006/jmbi.1997.0951.

[57] A. Bairoch, "PROSITE: A dictionary of sites and patterns in proteins," en, *Nucleic Acids Res.*, vol. 19 Suppl, no. Suppl, pp. 2241–2245, Apr. 1991, ISSN: 0305-1048. DOI: 10.1093/nar/19.suppl.2241.

[58] M. Källberg, H. Wang, S. Wang, *et al.*, "Template-based protein structure modeling using the RaptorX web server," en, *Nat. Protoc.*, vol. 7, no. 8, pp. 1511–1522, Jul. 2012, ISSN: 1754-2189,1750-2799. DOI: 10.1038/nprot.2012.085.

[59] J. Jumper, R. Evans, A. Pritzel, *et al.*, "Highly accurate protein structure prediction with AlphaFold," en, *Nature*, vol. 596, no. 7873, pp. 583–589, Aug. 2021, ISSN: 0028-0836,1476-4687. DOI: 10.1038/s41586-021-03819-2.

[60] H. M. Berman, T. Battistuz, T. N. Bhat, *et al.*, "The protein data bank," en, *Acta Crystallogr. D Biol. Crystallogr.*, vol. 58, no. Pt 61, pp. 899–907, Jun. 2002, ISSN: 0907-4449. DOI: 10.1107/s0907444902003451.

[61] J. M. Stuart, E. Segal, D. Koller, and S. K. Kim, "A gene-coexpression network for global discovery of conserved genetic modules," en, *Science*, vol. 302, no. 5643, pp. 249–255, Oct. 2003, ISSN: 0036-8075,1095-9203. DOI: 10.1126/science.1087447.

[62] R. Overbeek, M. Fonstein, M. D'Souza, G. D. Pusch, and N. Maltsev, "The use of gene clusters to infer functional coupling," en, *Proc. Natl. Acad. Sci. U. S. A.*, vol. 96, no. 6, pp. 2896–2901, Mar. 1999, ISSN: 0027-8424. DOI: 10.1073/pnas.96.6.2896.

[63] M. H. Medema, K. Blin, P. Cimermancic, *et al.*, "antiSMASH: Rapid identification, annotation and analysis of secondary metabolite biosynthesis gene clusters in bacterial and fungal genome sequences," en, *Nucleic Acids Res.*, vol. 39, no. Web Server issue, W339–46, Jul. 2011, ISSN: 0305-1048,1362-4962. DOI: 10.1093/nar/gkr466.

[64] M. Kanehisa, M. Araki, S. Goto, *et al.*, "KEGG for linking genomes to life and the environment," en, *Nucleic Acids Res.*, vol. 36, no. Database issue, pp. D480–4, Jan. 2008, ISSN: 0305-1048,1362-4962. DOI: 10.1093/nar/gkm882.

[65] R. D. Finn, A. Bateman, J. Clements, *et al.*, "Pfam: The protein families database," en, *Nucleic Acids Res.*, vol. 42, no. Database issue, pp. D222–30, Jan. 2014, ISSN: 0305-1048,1362-4962. DOI: 10.1093/nar/gkt1223.

[66] M. Ashburner, C. A. Ball, J. A. Blake, *et al.*, "Gene ontology: Tool for the unification of biology. the gene ontology consortium," en, *Nat. Genet.*, vol. 25, no. 1, pp. 25–29, May 2000, ISSN: 1061-4036. DOI: 10.1038/75556.

[67] M. Schena, D. Shalon, R. W. Davis, and P. O. Brown, "Quantitative monitoring of gene expression patterns with a complementary DNA microarray," en, *Science*, vol. 270, no. 5235, pp. 467–470, Oct. 1995, ISSN: 0036-8075,1095-9203. DOI: 10.1126/science.270.5235.467.

[68] G. Robertson, J. Schein, R. Chiu, *et al.*, "De novo assembly and analysis of RNA-seq data," en, *Nat. Methods*, vol. 7, no. 11, pp. 909–912, Nov. 2010, ISSN: 1548-7091,1548-7105. DOI: 10.1038/nmeth.1517.

[69] C. Trapnell, B. A. Williams, G. Pertea, *et al.*, "Transcript assembly and quantification by RNA-seq reveals unannotated transcripts and isoform switching during cell differentiation," en, *Nat. Biotechnol.*, vol. 28, no. 5, pp. 511–515, May 2010, ISSN: 1087-0156,1546-1696. DOI: 10.1038/nbt.1621.

[70] Y. Liao, G. K. Smyth, and W. Shi, "featureCounts: An efficient general purpose program for assigning sequence reads to genomic features," en, *Bioinformatics*, vol. 30, no. 7, pp. 923–930, Apr. 2014, ISSN: 1367-4803,1367-4811. DOI: 10.1093/bioinformatics/btt656.

[71] B. Li and C. N. Dewey, "RSEM: Accurate transcript quantification from RNA-seq data with or without a reference genome," en, *BMC Bioinformatics*, vol. 12, p. 323, Aug. 2011, ISSN: 1471-2105. DOI: 10.1186/1471-2105-12-323.

[72] J. Costa-Silva, D. Domingues, and F. M. Lopes, "RNA-seq differential expression analysis: An extended review and a software tool," en, *PLoS One*, vol. 12, no. 12, e0190152, Dec. 2017, ISSN: 1932-6203. DOI: 10.1371/journal.pone.0190152.

[73] M. D. Robinson, D. J. McCarthy, and G. K. Smyth, "edgeR: A bioconductor package for differential expression analysis of digital gene expression data," en, *Bioinformatics*, vol. 26, no. 1, pp. 139–140, Jan. 2010, ISSN: 1367-4803,1367-4811. DOI: 10.1093/bioinformatics/btp616.

[74] Y. Benjamini and Y. Hochberg, "Controlling the false discovery rate: A practical and powerful approach to multiple testing," en, *J. R. Stat. Soc.*, vol. 57, no. 1, pp. 289–300, Jan. 1995, ISSN: 0035-9246. DOI: 10.1111/j.2517-6161.1995.tb02031.x.

[75] A. Ben-Dor, R. Shamir, and Z. Yakhini, "Clustering gene expression patterns," en, *J. Comput. Biol.*, vol. 6, no. 3-4, pp. 281–297, 1999, ISSN: 1066-5277. DOI: 10.1089/106652799318274.

[76] B. Zhang and S. Horvath, "A general framework for weighted gene co-expression network analysis," en, *Stat. Appl. Genet. Mol. Biol.*, vol. 4, Article17, Aug. 2005, ISSN: 1544-6115. DOI: 10.2202/1544-6115.1128.

[77] P. D. Thomas, M. J. Campbell, A. Kejariwal, *et al.*, "PANTHER: A library of protein families and subfamilies indexed by function," en, *Genome Res.*, vol. 13, no. 9, pp. 2129–2141, Sep. 2003, ISSN: 1088-9051. DOI: 10.1101/gr.772403.

[78] U. Raudvere, L. Kolberg, I. Kuzmin, *et al.*, "G:profiler: A web server for functional enrichment analysis and conversions of gene lists (2019 update)," en, *Nucleic Acids Res.*, vol. 47, no. W1, W191–W198, Jul. 2019, ISSN: 0305-1048,1362-4962. DOI: 10.1093/nar/gkz369.

[79] A. Subramanian, P. Tamayo, V. K. Mootha, *et al.*, "Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles," en, *Proc. Natl. Acad. Sci. U. S. A.*, vol. 102, no. 43, pp. 15 545–15 550, Oct. 2005, ISSN: 0027-8424. DOI: 10.1073/pnas.0506580102.

[80] J. J. Goeman and P. Bühlmann, "Analyzing gene expression data in terms of gene sets: Methodological issues," en, *Bioinformatics*, vol. 23, no. 8, pp. 980–987, Apr. 2007, ISSN: 1367-4803,1367-4811. DOI: 10.1093/bioinformatics/btm051.

[81] S. Tyanova, T. Temu, and J. Cox, "The MaxQuant computational platform for mass spectrometry-based shotgun proteomics," *Nat. Protoc.*, vol. 11, pp. 2301–2319, Oct. 2016, ISSN: 1754-2189,1750-2799. DOI: 10.1038/nprot.2016.136.

[82] M. Ghoul and S. Mitri, "The ecology and evolution of microbial competition," en, *Trends Microbiol.*, vol. 24, no. 10, pp. 833–845, Oct. 2016, ISSN: 0966-842X,1878-4380. DOI: 10.1016/j.tim.2016.06.011.

[83] M. A. Wörheide, J. Krumsiek, G. Kastenmüller, and M. Arnold, "Multi-omics integration in biomedical research - a metabolomics-centric review," en, *Anal. Chim. Acta*, vol. 1141, pp. 144–162, Jan. 2021, ISSN: 0003-2670,1873-4324. DOI: 10.1016/j.aca.2020.10.038.

[84] H. Sharifi-Noghabi, O. Zolotareva, C. C. Collins, and M. Ester, "MOLI: Multi-omics late integration with deep neural

networks for drug response prediction," en, *Bioinformatics*, vol. 35, no. 14, pp. i501–i509, Jul. 2019, ISSN: 1367-4803,1367-4811. DOI: 10.1093/bioinformatics/btz318.

[85] S. Rendgen, *History of information graphics*, en, J. Wiedemann, Ed. Köln, Germany: Taschen, May 2019, ISBN: 9783836567671.

[86] F. Hutmacher, "Why is there so much more research on vision than on any other sensory modality?" en, *Front. Psychol.*, vol. 10, p. 2246, Oct. 2019, ISSN: 1664-1078. DOI: 10.3389/fpsyg.2019.02246.

[87] J. L. McClelland, "Phenomenology of perception," en, *Science*, vol. 201, no. 4359, pp. 899–900, Sep. 1978, ISSN: 0036-8075. DOI: 10.1126/science.201.4359.899-a.

[88] T. Nørretranders, *The user illusion: Cutting consciousness down to size*. London, England: Viking, Apr. 1998, ISBN: 9780670875795.

[89] M. McLuhan, W. Terrence Gordon, E. Lamberti, and D. Scheffel-Dunand, *The Gutenberg Galaxy: The Making of Typographic Man*, en. University of Toronto Press, Jan. 2011, ISBN: 9781442612693.

[90] C. Ware, *Visual Thinking for Design*, en. Elsevier, Jul. 2010, ISBN: 9780080558417.

[91] M. P. Mattson, "Superior pattern processing is the essence of the evolved human brain," en, *Front. Neurosci.*, vol. 8, p. 265, Aug. 2014, ISSN: 1662-4548,1662-453X. DOI: 10.3389/fnins.2014.00265.

[92] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer New York, 2006.

[93] T. Munzner, *Visualization Analysis and Design*, en. CRC Press, Dec. 2014, ISBN: 9781466508934.

[94] W. S. Cleveland and R. McGill, "Graphical perception: Theory, experimentation, and application to the development of graphical methods," *J. Am. Stat. Assoc.*, vol. 79, no. 387, pp. 531–554, Sep. 1984, ISSN: 0162-1459. DOI: 10.1080/01621459.1984.10478080.

[95] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," in *Proceedings 1996 IEEE Symposium on Visual Languages*, Boulder, CO, USA:

IEEE, 1996, pp. 336–343, ISBN: 9780818675089. DOI: 10.1109/VL.1996.545307.

[96] E. Tufte, "The visual display of quantitative information," *Technometrics*, vol. 44, pp. 400–400, Jan. 1985, ISSN: 0040-1706,1537-2723. DOI: 10.1198/tech.2002.s78.

[97] O. Inbar, N. Tractinsky, and J. Meyer, "Minimalism in information visualization: Attitudes towards maximizing the data-ink ratio," in *Proceedings of the 14th European conference on Cognitive ergonomics: invent! explore!*, ser. ECCE '07, New York, NY, USA: Association for Computing Machinery, Aug. 2007, pp. 185–188, ISBN: 9781847998491. DOI: 10.1145/1362550.1362587.

[98] S. Bateman, R. L. Mandryk, C. Gutwin, A. Genest, D. McDine, and C. Brooks, "Useful junk? the effects of visual embellishment on comprehension and memorability of charts," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '10, New York, NY, USA: Association for Computing Machinery, Apr. 2010, pp. 2573–2582, ISBN: 9781605589299. DOI: 10.1145/1753326.1753716.

[99] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," en, *Proc. Natl. Acad. Sci. U. S. A.*, vol. 95, no. 25, pp. 14 863–14 868, Dec. 1998, ISSN: 0027-8424. DOI: 10.1073/pnas.95.25.14863.

[100] A. Lex, N. Gehlenborg, H. Strobelt, R. Vuillemot, and H. Pfister, "UpSet: Visualization of intersecting sets," en, *IEEE Trans. Vis. Comput. Graph.*, vol. 20, no. 12, pp. 1983–1992, Dec. 2014, ISSN: 1077-2626,1941-0506. DOI: 10.1109/TVCG.2014.2346248.

[101] M. Brehmer and T. Munzner, "A multi-level typology of abstract visualization tasks," en, *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 12, pp. 2376–2385, Dec. 2013, ISSN: 1077-2626,1941-0506. DOI: 10.1109/TVCG.2013.124.

[102] D. Fisher and M. Meyer, *Making Data Visual: A Practical Guide to Using Visualization for Insight*, en. "O'Reilly Media, Inc.", Dec. 2017, ISBN: 9781491928448.

[103] M. J. Goldman, B. Craft, M. Hastie, *et al.*, "Visualizing and interpreting cancer genomics data via the xena platform," en, *Nat. Biotechnol.*, vol. 38, no. 6, pp. 675–678, Jun. 2020, ISSN: 1087-0156,1546-1696. DOI: 10.1038/s41587-020-0546-8.

[104] E. Cerami, J. Gao, U. Dogrusoz, *et al.*, "The cBio cancer genomics portal: An open platform for exploring multidimensional cancer genomics data," en, *Cancer Discov.*, vol. 2, no. 5, pp. 401–404, May 2012, ISSN: 2159-8274,2159-8290. DOI: 10.1158/2159-8290.CD-12-0095.

[105] J. Gao, B. A. Aksoy, U. Dogrusoz, *et al.*, "Integrative analysis of complex cancer genomics and clinical profiles using the cBioPortal," en, *Sci. Signal.*, vol. 6, no. 269, p. l1, Apr. 2013, ISSN: 1937-9145,1945-0877. DOI: 10.1126/scisignal.2004088.

[106] C. Bertelli, M. R. Laird, K. P. Williams, *et al.*, "IslandViewer 4: Expanded prediction of genomic islands for larger-scale datasets," en, *Nucleic Acids Res.*, vol. 45, no. W1, W30–W35, Jul. 2017, ISSN: 0305-1048,1362-4962. DOI: 10.1093/nar/gkx343.

[107] J. Botas, Á. Rodríguez Del Río, J. Giner-Lamia, and J. Huerta-Cepas, "GeCoViz: Genomic context visualisation of prokaryotic genes from a functional and evolutionary perspective," en, *Nucleic Acids Res.*, vol. 50, no. W1, W352–W357, Jul. 2022, ISSN: 0305-1048,1362-4962. DOI: 10.1093/nar/gkac367.

[108] T. Carver, N. Thomson, A. Bleasby, M. Berriman, and J. Parkhill, "DNAPlotter: Circular and linear interactive genome visualization," en, *Bioinformatics*, vol. 25, no. 1, pp. 119–120, Jan. 2009, ISSN: 1367-4803,1367-4811. DOI: 10.1093/bioinformatics/btn578.

[109] L. Overmars, S. A. F. T. van Hijum, R. J. Siezen, and C. Francke, "CiVi: Circular genome visualization with unique features to analyze sequence elements," en, *Bioinformatics*, vol. 31, no. 17, pp. 2867–2869, Sep. 2015, ISSN: 1367-4803,1367-4811. DOI: 10.1093/bioinformatics/btv249.

[110] P. Kerpedjiev, N. Abdennur, F. Lekschas, *et al.*, "HiGlass: Web-based visual exploration and analysis of genome interaction maps," en, *Genome Biol.*, vol. 19, no. 1, p. 125, Aug. 2018, ISSN: 1465-6906. DOI: 10.1186/s13059-018-1486-1.

[111] M. Meyer, T. Munzner, and H. Pfister, "MizBee: A multiscale synteny browser," en, *IEEE Trans. Vis. Comput. Graph.*, vol. 15, no. 6, pp. 897–904, 2009, ISSN: 1077-2626. DOI: 10.1109/TVCG.2009.167.

[112] D. J. Newman and G. M. Cragg, "Natural products as sources of new drugs from 1981 to 2014," en, *J. Nat. Prod.*, vol. 79, no. 3, pp. 629–661, Mar. 2016, ISSN: 0163-3864,1520-6025. DOI: 10.1021/acs.jnatprod.5b01055.

[113] Y. Yan, Q. Liu, S. E. Jacobsen, and Y. Tang, "The impact and prospect of natural product discovery in agriculture: New technologies to explore the diversity of secondary metabolites in plants and microorganisms for applications in agriculture," en, *EMBO Rep.*, vol. 19, no. 11, Nov. 2018, ISSN: 1469-221X,1469-3178. DOI: 10.15252/embr.201846824.

[114] D. Mao, B. K. Okada, Y. Wu, F. Xu, and M. R. Seyedsayamdost, "Recent advances in activating silent biosynthetic gene clusters in bacteria," en, *Curr. Opin. Microbiol.*, vol. 45, pp. 156–163, Oct. 2018, ISSN: 1369-5274,1879-0364. DOI: 10.1016/j.mib.2018.05.001.

[115] K. Blin, S. Shaw, H. E. Augustijn, *et al.*, "antiSMASH 7.0: New and improved predictions for detection, regulation, chemical structures and visualisation," en, *Nucleic Acids Res.*, vol. 51, no. W1, W46–W50, Jul. 2023, ISSN: 0305-1048,1362-4962. DOI: 10.1093/nar/gkad344.

[116] K. Palaniappan, I.-M. A. Chen, K. Chu, *et al.*, "IMG-ABC v.5.0: An update to the IMG/atlas of biosynthetic gene clusters knowledgebase," en, *Nucleic Acids Res.*, vol. 48, no. D1, pp. D422–D430, Jan. 2020, ISSN: 0305-1048,1362-4962. DOI: 10.1093/nar/gkz932.

[117] R. J. S. Baerends, W. K. Smits, A. de Jong, L. W. Hamoen, J. Kok, and O. P. Kuipers, "Genome2D: A visualization tool for the rapid analysis of bacterial transcriptome data," en, *Genome Biol.*, vol. 5, no. 5, R37, Apr. 2004, ISSN: 1465-6906. DOI: 10.1186/gb-2004-5-5-r37.

[118] R. W. W. Brouwer, S. A. F. T. van Hijum, and O. P. Kuipers, "MINOMICS: Visualizing prokaryote transcriptomics and proteomics data in a genomic context," en, *Bioinformatics*, vol. 25, no. 1, pp. 139–140, Jan. 2009, ISSN: 1367-4803,1367-4811. DOI: 10.1093/bioinformatics/btn588.

[119] N. Lee, W. Kim, J. Chung, *et al.*, "Iron competition triggers antibiotic biosynthesis in streptomyces coelicolor during coculture with myxococcus xanthus," en, *ISME J.*, vol. 14, no. 5, pp. 1111–1124, May 2020, ISSN: 1751-7362,1751-7370. DOI: 10.1038/s41396-020-0594-6.

[120] M. Krzywinski, J. Schein, I. Birol, *et al.*, "Circos: An information aesthetic for comparative genomics," en, *Genome Res.*, vol. 19, no. 9, pp. 1639–1645, Sep. 2009, ISSN: 1088-9051,1549-5469. DOI: 10.1101/gr.092759.109.

[121] T. Manz, S. L'Yi, and N. Gehlenborg, "Gos: A declarative library for interactive genomics visualization in python," en, *Bioinformatics*, vol. 39, no. 1, Jan. 2023, ISSN: 1367-4803,1367-4811. DOI: 10.1093/bioinformatics/btad050.

[122] A. Satyanarayan, R. Russell, J. Hoffswell, and J. Heer, "Reactive vega: A streaming dataflow architecture for declarative interactive visualization," en, *IEEE Trans. Vis. Comput. Graph.*, vol. 22, no. 1, pp. 659–668, Jan. 2016, ISSN: 1077-2626,1941-0506. DOI: 10.1109/TVCG.2015.2467091.

[123] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer, "Vega-lite: A grammar of interactive graphics," en, *IEEE Trans. Vis. Comput. Graph.*, vol. 23, no. 1, pp. 341–350, Jan. 2017, ISSN: 1077-2626,1941-0506. DOI: 10.1109/TVCG.2016.2599030.

[124] K. Lavikka, J. Oikkonen, R. Lehtonen, J. Hynninen, S. Hietanen, and S. Hautaniemi, "GenomeSpy: Grammar-based interactive genome visualization," *F1000Res.*, vol. 9, Aug. 2020. DOI: 10.7490/F1000RESEARCH.1118237.1.

[125] K. Lavikka, J. Oikkonen, Y. Li, *et al.*, "Deciphering cancer genomes with GenomeSpy: A grammar-based visualization toolkit," en, *bioRxiv*, p. 2023.10.06.561159, Oct. 2023. DOI: 10.1101/2023.10.06.561159.

[126] P. S. Garcia, F. Jauffrit, C. Grangeasse, and C. Brochier-Armanet, "GeneSpy, a user-friendly and flexible genomic context visualizer," en, *Bioinformatics*, vol. 35, no. 2, pp. 329–331, Jan. 2019, ISSN: 1367-4803,1367-4811. DOI: 10.1093/bioinformatics/bty459.

[127] A. Cumsille, R. E. Durán, A. Rodríguez-Delherbe, *et al.*, "GenoVi, an open-source automated circular genome visualizer for bacteria and archaea," en, *PLoS Comput. Biol.*, vol. 19, no. 4, e1010998, Apr. 2023, ISSN: 1553-734X,1553-7358. DOI: 10.1371/journal.pcbi.1010998.

[128] E. Ståhlbom, J. Molin, A. Ynnerman, and C. Lundström, "Should I make it round? suitability of circular and linear layouts for comparative tasks with matrix and connective data," en, *Comput. Graph. Forum*, vol. 43, no. 3, Jun. 2024, ISSN: 0167-7055,1467-8659. DOI: 10.1111/cgf.15102.

[129] N. Arora, V. J. Schuenemann, G. Jäger, *et al.*, "Origin of modern syphilis and emergence of a pandemic treponema pallidum cluster," en, *Nat Microbiol*, vol. 2, p. 16 245, Dec. 2016, ISSN: 2058-5276. DOI: 10.1038/nmicrobiol.2016.245.

[130] M. A. Harris, J. Clark, A. Ireland, *et al.*, "The gene ontology (GO) database and informatics resource," en, *Nucleic Acids Res.*, vol. 32, no. Database issue, pp. D258–61, Jan. 2004, ISSN: 0305-1048,1362-4962. DOI: 10.1093/nar/gkh036.

[131] D. V. Klopfenstein, L. Zhang, B. S. Pedersen, *et al.*, "GOATOOLS: A python library for gene ontology analyses," en, *Sci. Rep.*, vol. 8, no. 1, p. 10 872, Jul. 2018, ISSN: 2045-2322. DOI: 10.1038/s41598-018-28948-z.

[132] P. W. Lord, R. D. Stevens, A. Brass, and C. A. Goble, "Investigating semantic similarity measures across the gene ontology: The relationship between sequence and annotation," en, *Bioinformatics*, vol. 19, no. 10, pp. 1275–1283, Jul. 2003, ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btg153.

[133] A. Brionne, A. Juanchich, and C. Hennequet-Antier, "ViSEAGO: A bioconductor package for clustering biological functions using gene ontology and semantic similarity," en, *BioData Min.*, vol. 12, p. 16, Aug. 2019, ISSN: 1756-0381. DOI: 10.1186/s13040-019-0204-1.

[134] R. Kolde and J. Vilo, "GOsummaries: An R package for visual functional annotation of experimental data," en, *F1000Res.*, vol. 4, p. 574, Aug. 2015, ISSN: 2046-1402. DOI: 10.12688/f1000research.6925.1.

[135] F. Supek, M. Bošnjak, N. Škunca, and T. Šmuc, "REVIGO summarizes and visualizes long lists of gene ontology terms," en, *PLoS One*, vol. 6, no. 7, e21800, Jul. 2011, ISSN: 1932-6203. DOI: 10.1371/journal.pone.0021800.

[136] I. Kuznetsova, A. Lugmayr, S. J. Siira, O. Rackham, and A. Filipovska, "CirGO: An alternative circular way of visualising gene ontology terms," en, *BMC Bioinformatics*, vol. 20, no. 1, p. 84, Feb. 2019, ISSN: 1471-2105. DOI: 10.1186/s12859-019-2671-2.

[137] V. Fortino, H. Alenius, and D. Greco, "BACA: Bubble chArt to compare annotations," en, *BMC Bioinformatics*, vol. 16, p. 37, Feb. 2015, ISSN: 1471-2105. DOI: 10.1186/s12859-015-0477-4.

[138] P. Resnik, "Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language," en, *jair*, vol. 11, pp. 95–130, Jul. 1999, ISSN: 1076-9757,1076-9757. DOI: 10.1613/jair.514.

[139] D. Lin *et al.*, "An information-theoretic definition of similarity," in *Icml*, vol. 98, pdfs.semanticscholar.org, 1998, pp. 296–304.

[140] J. Z. Wang, Z. Du, R. Payattakool, P. S. Yu, and C.-F. Chen, "A new method to measure the semantic similarity of GO terms," en, *Bioinformatics*, vol. 23, no. 10, pp. 1274–1281, May 2007, ISSN: 1367-4803,1367-4811. DOI: 10.1093/bioinformatics/btm087.

[141] S. Sulheim, T. Kumelj, D. van Dissel, *et al.*, "Enzyme-constrained models and omics analysis of streptomyces coelicolor reveal metabolic changes that enhance heterologous production," en, *iScience*, vol. 23, no. 9, p. 101 525, Sep. 2020, ISSN: 2589-0042. DOI: 10.1016/j.isci.2020.101525.

[142] M. Bruls, K. Huizing, and J. J. van Wijk, "Squarified treemaps," in *Data Visualization 2000*, Springer Vienna, 2000, pp. 33–42. DOI: 10.1007/978-3-7091-6783-0_4.

[143] *Flask — A micro web framework written in Python*, https://flask.palletsprojects.com/, Accessed: 2025-2-2, 2010.

[144] Facebook, *React - A JavaScript library for building user interfaces*, https://reactjs.org/, Accessed: 2025-2-2, 2013.

[145] *MobX — simple, scalable state management*, https://mobx.js.org/, Accessed: 2025-2-2, 2015.

[146] M. Bostock, V. Ogievetsky, and J. Heer, "D^3: Data-driven documents," en, *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 12, pp. 2301–2309, Dec. 2011, ISSN: 1077-2626,1941-0506. DOI: 10.1109/TVCG.2011.185.

[147] A. Lavelle, T. W. Hoffmann, H.-P. Pham, P. Langella, E. Guédon, and H. Sokol, "Baseline microbiota composition modulates antibiotic-mediated effects on the gut microbiota and host," en, *Microbiome*, vol. 7, no. 1, p. 111, Aug. 2019, ISSN: 2049-2618. DOI: 10.1186/s40168-019-0725-3.

[148] M. E. Ritchie, B. Phipson, D. Wu, *et al.*, "Limma powers differential expression analyses for RNA-sequencing and microarray studies," en, *Nucleic Acids Res.*, vol. 43, no. 7, e47, Apr. 2015, ISSN: 0305-1048,1362-4962. DOI: 10.1093/nar/gkv007.

[149] A. E. Singh and B. Romanowski, "Syphilis: Review with emphasis on clinical, epidemiologic, and some biologic features," en, *Clin. Microbiol. Rev.*, vol. 12, no. 2, pp. 187–209, Apr. 1999, ISSN: 0893-8512. DOI: 10.1128/CMR.12.2.187.

[150] M. Pla-Díaz, L. Sánchez-Busó, L. Giacani, *et al.*, "Evolutionary processes in the emergence and recent spread of the syphilis agent, treponema pallidum," en, *Mol. Biol. Evol.*, vol. 39, no. 1, Jan. 2022, ISSN: 0737-4038,1537-1719. DOI: 10.1093/molbev/msab318.

[151] J. Brooke, *SUS -a quick and dirty usability scale*, http://www.tbistafftraining.info/smartphones/documents/b5_during_the_trial_usability_scale_v1_09aug11.pdf, Accessed: 2023-3-15, 1996.

[152] R. Hernández-de-Diego, S. Tarazona, C. Martínez-Mira, *et al.*, "PaintOmics 3: A web resource for the pathway analysis and visualization of multi-omics data," en, *Nucleic Acids Res.*, vol. 46, no. W1, W503–W509, Jul. 2018, ISSN: 0305-1048,1362-4962. DOI: 10.1093/nar/gky466.

[153] A. Inselberg, "The plane with parallel coordinates," *Vis. Comput.*, vol. 1, no. 2, pp. 69–91, Aug. 1985, ISSN: 0178-2789,1432-2315. DOI: 10.1007/BF01898350.

[154] A. Lex, M. Streit, H.-J. Schulz, *et al.*, "StratomeX: Visual analysis of large-scale heterogeneous genomics data for cancer subtype characterization," en, *Comput. Graph. Forum*, vol. 31, no. 33, pp. 1175–1184, Jun. 2012, ISSN: 0167-7055. DOI: 10.1111/j.1467-8659.2012.03110.x.

[155] M. Bersanelli, E. Mosca, D. Remondini, *et al.*, "Methods for the integration of multi-omics data: Mathematical aspects," en, *BMC Bioinformatics*, vol. 17 Suppl 2, no. Suppl 2, p. 15, Jan. 2016, ISSN: 1471-2105. DOI: 10.1186/s12859-015-0857-9.

[156] S. Huang, K. Chaudhary, and L. X. Garmire, "More is better: Recent progress in multi-omics data integration methods," en, *Front. Genet.*, vol. 8, p. 84, Jun. 2017, ISSN: 1664-8021. DOI: 10.3389/fgene.2017.00084.

[157] N. Rappoport and R. Shamir, "Multi-omic and multi-view clustering algorithms: Review and cancer benchmark," en, *Nucleic Acids Res.*, vol. 47, no. 2, p. 1044, Jan. 2019, ISSN: 0305-1048,1362-4962. DOI: 10.1093/nar/gky1226.

[158] G. Tini, L. Marchetti, C. Priami, and M.-P. Scott-Boyer, "Multi-omics integration - a comparison of unsupervised clustering methodologies," *Brief. Bioinform.*, Jul. 2019, ISSN: 1467-5463. DOI: 10.1093/bib/bbx167.

[159] S. Ghosh, A. Datta, and H. Choi, "multiSLIDE: A web server for exploring connected elements of biological pathways in multi-omics data," *bioRxiv*, Oct. 2019. DOI: 10.1101/812271.

[160] N. Gehlenborg, "Integrating data: Different analytical tasks require different visual representations," *Nat. Methods*, vol. 9, p. 315, Apr. 2012, ISSN: 1548-7091.

[161] J. Vercruysse, M. Van Bel, C. M. Osuna-Cruz, *et al.*, "Comparative transcriptomics enables the identification of functional orthologous genes involved in early leaf growth," en, *Plant Biotechnol. J.*, vol. 18, no. 2, pp. 553–567, Feb. 2020, ISSN: 1467-7644,1467-7652. DOI: 10.1111/pbi.13223.

[162] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding," *Symposium on Discrete Algorithms*, pp. 1027–1035, Jan. 2007.

[163] H. Mi, D. Ebert, A. Muruganujan, *et al.*, "PANTHER version 16: A revised family classification, tree-based classification tool, enhancer regions and extensive API," en, *Nucleic Acids Res.*, vol. 49, no. D1, pp. D394–D403, Jan. 2021, ISSN: 0305-1048,1362-4962. DOI: 10.1093/nar/gkaa1106.

[164] M. Grinberg, *Flask Web Development*, en. "O'Reilly Media, Inc.", Mar. 2018, ISBN: 9781491991695.

[165] A. J. Hoogendijk, F. Pourfarzad, C. E. M. Aarts, *et al.*, "Dynamic transcriptome-proteome correlation networks reveal human myeloid differentiation and neutrophil-specific programming," en, *Cell Rep.*, vol. 29, no. 8, 2505–2519.e4, Nov. 2019, ISSN: 2211-1247. DOI: 10.1016/j.celrep.2019.10.082.

[166] J. P. Gomez-Escribano and M. J. Bibb, "Engineering streptomyces coelicolor for heterologous expression of secondary metabolite gene clusters," en, *Microb. Biotechnol.*, vol. 4, no. 2, pp. 207–215, Mar. 2011, ISSN: 1751-7915. DOI: 10.1111/j.1751-7915.2010.00219.x.

[167] J. F. Martín, A. Rodríguez-García, and P. Liras, "The master regulator PhoP coordinates phosphate and nitrogen metabolism, respiration, cell differentiation and antibiotic biosynthesis: Comparison in streptomyces coelicolor and streptomyces avermitilis," en, *J. Antibiot.*, vol. 70, no. 5, pp. 534–541, Mar. 2017, ISSN: 0021-8820. DOI: 10.1038/ja.2017.19.

[168] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," *KDD*, pp. 226–231, Aug. 1996, ISSN: 2154-817X.

[169] R. Shen, A. B. Olshen, and M. Ladanyi, "Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis," en, *Bioinformatics*, vol. 25, no. 22, pp. 2906–2912, Nov. 2009, ISSN: 1367-4803,1367-4811. DOI: 10.1093/bioinformatics/btp543.

[170] M. Kern, A. Lex, N. Gehlenborg, and C. R. Johnson, "Interactive visual exploration and refinement of cluster assignments," en, *BMC Bioinformatics*, vol. 18, no. 1, p. 406, Sep. 2017, ISSN: 1471-2105. DOI: 10.1186/s12859-017-1813-7.

[171] M. Streit, A. Lex, S. Gratzl, *et al.*, "Guided visual exploration of genomic stratifications in cancer," en, *Nat. Methods*, vol. 11, no. 9, pp. 884–885, Sep. 2014, ISSN: 1548-7091,1548-7105. DOI: 10.1038/nmeth.3088.

[172] H. X. Dang, B. S. White, S. M. Foltz, *et al.*, "ClonEvol: Clonal ordering and visualization in cancer sequencing," en, *Ann. Oncol.*, vol. 28, no. 12, pp. 3076–3082, Dec. 2017, ISSN: 0923-7534,1569-8041. DOI: 10.1093/annonc/mdx517.

[173] C. A. Miller, J. McMichael, H. X. Dang, *et al.*, "Visualizing tumor evolution with the fishplot package for R," en, *BMC Genomics*, vol. 17, no. 1, p. 880, Nov. 2016, ISSN: 1471-2164. DOI: 10.1186/s12864-016-3195-z.

[174] M. Monroe, R. Lan, H. Lee, C. Plaisant, and B. Shneiderman, "Temporal event sequence simplification," en, *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 12, pp. 2227–2236, Dec. 2013, ISSN: 1077-2626,1941-0506. DOI: 10.1109/TVCG.2013.200.

[175] D. Gotz and H. Stavropoulos, "DecisionFlow: Visual analytics for high-dimensional temporal event sequence data," en, *IEEE Trans. Vis. Comput. Graph.*, vol. 20, no. 12, pp. 1783–1792, Dec. 2014, ISSN: 1077-2626,1941-0506. DOI: 10.1109/TVCG.2014.2346682.

[176] A. Perer and J. Sun, "MatrixFlow: Temporal network visual analytics to track symptom evolution during disease progression," en, *AMIA Annu. Symp. Proc.*, vol. 2012, pp. 716–725, Nov. 2012, ISSN: 1942-597X,1559-4076.

[177] S. Malik, F. Du, M. Monroe, E. Onukwugha, C. Plaisant, and B. Shneiderman, "Cohort comparison of event sequences with balanced integration of visual analytics and statistics," in *Proceedings of the 20th International Conference on Intelligent User Interfaces*, ser. IUI '15, New York, NY, USA: Association for Computing Machinery, Mar. 2015, pp. 38–49, ISBN: 9781450333061. DOI: 10.1145/2678025.2701407.

[178] S. Gratzl, N. Gehlenborg, A. Lex, H. Pfister, and M. Streit, "Domino: Extracting, comparing, and manipulating subsets across multiple tabular datasets," en, *IEEE Trans. Vis. Comput. Graph.*, vol. 20, no. 12, pp. 2023–2032, Dec. 2014, ISSN: 1077-2626,1941-0506. DOI: 10.1109/TVCG.2014.2346260.

[179] J. Knapp, J. Zeratsky, and B. Kowitz, *Sprint: How to Solve Big Problems and Test New Ideas in Just Five Days*, en. Simon and Schuster, Mar. 2016, ISBN: 9781501121777.

[180] S. Gratzl, A. Lex, N. Gehlenborg, H. Pfister, and M. Streit, "LineUp: Visual analysis of multi-attribute rankings," en, *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 12, pp. 2277–2286, Dec. 2013, ISSN: 1077-2626,1941-0506. DOI: 10.1109/TVCG.2013.173.

[181] A. R. Wilcox, "Indices of qualitative variation and political measurement," *West. Polit. Q.*, vol. 26, no. 2, pp. 325–343, Jun. 1973, ISSN: 0043-4078. DOI: 10.1177/106591297302600209.

[182] G. D. Kader and M. Perry, "Variability for categorical variables," *J. Stat. Educ.*, vol. 15, no. 2, Jul. 2007. DOI: 10.1080/10691898.2007.11889465.

[183] B. E. Johnson, T. Mazor, C. Hong, *et al.*, "Mutational analysis reveals the origin and therapy-driven evolution of recurrent glioma," en, *Science*, vol. 343, no. 6167, pp. 189–193, Jan. 2014, ISSN: 0036-8075,1095-9203. DOI: 10.1126/science.1239947.

[184] L. Liu and S. L. Gerson, "Targeted modulation of MGMT: Clinical implications," en, *Clin. Cancer Res.*, vol. 12, no. 2, pp. 328–331, Jan. 2006, ISSN: 1078-0432. DOI: 10.1158/1078-0432.CCR-05-2543.

[185] A. Evren and E. Ustaoğlu, "Measures of qualitative variation in the case of maximum entropy," en, *Entropy*, vol. 19, no. 5, p. 204, May 2017, ISSN: 1099-4300,1099-4300. DOI: 10.3390/e19050204.

[186] S. Gratzl, A. Lex, N. Gehlenborg, N. Cosgrove, and M. Streit, "From visual exploration to storytelling and back again," en, *Comput. Graph. Forum*, vol. 35, no. 3, pp. 491–500, Jun. 2016, ISSN: 0167-7055. DOI: 10.1111/cgf.12925.

[187] Z. Cutler, K. Gadhave, and A. Lex, "Trrack: A library for provenance-tracking in web-based visualizations," in *2020 IEEE Visualization Conference (VIS)*, IEEE, Oct. 2020, pp. 116–120, ISBN: 9781728180144,9781728180151. DOI: 10.1109/VIS47514.2020.00030.

[188] P. Saraiya, C. North, and K. Duca, "An insight-based methodology for evaluating bioinformatics visualizations," en, *IEEE Trans. Vis. Comput. Graph.*, vol. 11, no. 4, pp. 443–456, Jul. 2005, ISSN: 1077-2626. DOI: 10.1109/TVCG.2005.53.

[189] S. L. Milgram, "The small world problem," *Psychol. Today*, vol. 2, no. 1, pp. 60–67, 1967, ISSN: 0033-3107. DOI: 10.1037/e400002009-005.

[190] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," en, *Nature*, vol. 393, no. 6684, pp. 440–442, Jun. 1998, ISSN: 0028-0836,1476-4687. DOI: 10.1038/30918.

[191] F. Karinthy, "Láncszemek," *Minden másképp van*, 1929.

[192] M. Newman, A.-L. Barabási, and D. J. Watts, *The Structure and Dynamics of Networks* (Princeton Studies in Complexity), en. Princeton University Press, Oct. 2011, vol. 12, ISBN: 9781400841356. DOI: 10.1515/9781400841356.

[193] A.-L. Barabási and Z. N. Oltvai, "Network biology: Understanding the cell's functional organization," en, *Nat. Rev. Genet.*, vol. 5, no. 2, pp. 101–113, Feb. 2004, ISSN: 1471-0056,1471-0064. DOI: 10.1038/nrg1272.

[194] V. Filipov, A. Arleo, and S. Miksch, "Are we there yet? a roadmap of network visualization from surveys to task taxonomies," en, *Comput. Graph. Forum*, vol. 42, no. 6, e14794, Apr. 2023, ISSN: 0167-7055,1467-8659. DOI: 10.1111/cgf.14794.

[195] A. Nocaj, M. Ortmann, and U. Brandes, "Untangling hairballs," in *Graph Drawing*, E. Bayro-Corrochano and E. Hancock, Eds., ser. Lecture Notes in Computer Science, vol. 8827, Cham: Springer Berlin Heidelberg, 2014, pp. 101–112, ISBN: 9783319125671,9783319125688. DOI: 10.1007/978-3-662-45803-7_9.

[196] J. H. Fowler and N. A. Christakis, "Dynamic spread of happiness in a large social network: Longitudinal analysis over 20 years in the framingham heart study," en, *BMJ*, vol. 337, no. dec04 2, a2338, Dec. 2008, ISSN: 0959-8138,1756-1833. DOI: 10.1136/bmj.a2338.

[197] M. Pellizzari, "Do Friends and Relatives Really Help in Getting a Good Job?" *ILR Review*, vol. 63, no. 3, pp. 494–510, Apr. 2010, ISSN: 0019-7939,2162-271X. DOI: 10.1177/001979391006300307.

[198] I. A. Kovács, K. Luck, K. Spirohn, *et al.*, "Network-based prediction of protein interactions," en, *Nat. Commun.*, vol. 10, no. 1, p. 1240, Mar. 2019, ISSN: 2041-1723. DOI: 10.1038/s41467-019-09177-y.

[199] K. Xu, I. Bezakova, L. Bunimovich, and S. V. Yi, "Path lengths in protein-protein interaction networks and biological complexity," en, *Proteomics*, vol. 11, no. 10, pp. 1857–1867, May 2011, ISSN: 1615-9853,1615-9861. DOI: 10.1002/pmic.201000684.

[200] W. Ali, T. Rito, G. Reinert, F. Sun, and C. M. Deane, "Alignment-free protein interaction network comparison," en, *Bioinformatics*, vol. 30, no. 17, pp. i430–7, Sep. 2014, ISSN: 1367-4803,1367-4811. DOI: 10.1093/bioinformatics/btu447.

[201] M. Spreen, "Sampling personal network structures: Statistical inference in ego-graphs," Ph.D. dissertation, s.n. / University of Groningen, 1999.

[202] J. Zhao, M. Glueck, F. Chevalier, Y. Wu, and A. Khan, "Egocentric Analysis of Dynamic Networks with EgoLines," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, ser. CHI '16, New York, NY, USA: Association for Computing Machinery, May 2016, pp. 5003–5014, ISBN: 9781450333627. DOI: 10.1145/2858036.2858488.

[203] *Bio+MedVis challenge @ IEEE VIS*, http://biovis.net/2023/biovisChallenges_vis/, Accessed: 2025-2-2, 2023.

[204] E. Gonçalves, R. C. Poulos, Z. Cai, *et al.*, "Pan-cancer proteomic map of 949 human cell lines," en, *Cancer Cell*, vol. 40,

no. 8, 835–849.e8, Aug. 2022, ISSN: 1535-6108,1878-3686. DOI: 10.1016/j.ccell.2022.06.010.

[205] C. Vehlow, F. Beck, and D. Weiskopf, "Visualizing group structures in graphs: A survey," en, *Comput. Graph. Forum*, vol. 36, no. 6, pp. 201–225, Sep. 2017, ISSN: 0167-7055,1467-8659. DOI: 10.1111/cgf.12872.

[206] P. Isenberg, F. Heimerl, S. Koch, *et al.*, "Vispubdata.org: A metadata collection about IEEE visualization (VIS) publications," en, *IEEE Trans. Vis. Comput. Graph.*, vol. 23, no. 9, pp. 2199–2206, Sep. 2017, ISSN: 1077-2626,1941-0506. DOI: 10.1109/TVCG.2016.2615308.

[207] C. Nobre, M. Meyer, M. Streit, and A. Lex, "The state of the art in visualizing multivariate networks," en, *Comput. Graph. Forum*, vol. 38, no. 3, pp. 807–832, Jun. 2019, ISSN: 0167-7055,1467-8659. DOI: 10.1111/cgf.13728.

[208] P. Shannon, A. Markiel, O. Ozier, *et al.*, "Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks," *Genome Res.*, vol. 13, no. 11, pp. 2498–2504, Nov. 2003, ISSN: 1088-9051. DOI: 10/b7kgpg.

[209] C. D. Stolper, M. Kahng, Z. Lin, *et al.*, "GLO-STIX: Graph-Level Operations for Specifying Techniques and Interactive eXploration," en, *IEEE Trans. Vis. Comput. Graph.*, vol. 20, no. 12, pp. 2320–2328, Dec. 2014, ISSN: 1077-2626,1941-0506. DOI: 10.1109/TVCG.2014.2346444.

[210] D. Archambault, T. Munzner, and D. Auber, "TopoLayout: Multilevel graph layout by topological features," en, *IEEE Trans. Vis. Comput. Graph.*, vol. 13, no. 2, pp. 305–317, 2007, ISSN: 1077-2626. DOI: 10.1109/TVCG.2007.46.

[211] S. van den Elzen and J. J. van Wijk, "Multivariate Network Exploration and Presentation: From Detail to Overview via Selections and Aggregations," en, *IEEE Trans. Vis. Comput. Graph.*, vol. 20, no. 12, pp. 2310–2319, Dec. 2014, ISSN: 1077-2626,1941-0506. DOI: 10.1109/TVCG.2014.2346441.

[212] N. Henry, J.-D. Fekete, and M. J. McGuffin, "NodeTrix: A hybrid visualization of social networks," en, *IEEE Trans. Vis. Comput. Graph.*, vol. 13, no. 6, pp. 1302–1309, 2007, ISSN: 1077-2626. DOI: 10.1109/TVCG.2007.70582.

[213] L. Angori, W. Didimo, F. Montecchiani, D. Pagliuca, and A. Tappini, "ChordLink: A New Hybrid Visualization Model," in *Lecture Notes in Computer Science*, ser. Lecture notes in computer science, D. Archambault and C. D. Tóth, Eds., vol. 11904, Cham: Springer International Publishing, 2019, pp. 276–290, ISBN: 9783030358013,9783030358020. DOI: 10.1007/978-3-030-35802-0_22.

[214] M. Burch, C. Vehlow, N. Konevtsova, and D. Weiskopf, "Evaluating partially drawn links for directed graph edges," in *Graph Drawing*, ser. Lecture notes in computer science, M. Van Kreveld and B. Speckmann, Eds., vol. 7034, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 226–237, ISBN: 9783642258770,9783642258787. DOI: 10.1007/978-3-642-25878-7_22.

[215] B. Shneiderman and A. Aris, "Network visualization by semantic substrates," en, *IEEE Trans. Vis. Comput. Graph.*, vol. 12, no. 5, pp. 733–740, 2006, ISSN: 1077-2626. DOI: 10.1109/TVCG.2006.166.

[216] D. Holten, "Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data," en, *IEEE Trans. Vis. Comput. Graph.*, vol. 12, no. 5, pp. 741–748, 2006, ISSN: 1077-2626. DOI: 10.1109/TVCG.2006.147.

[217] H. Zhou, P. Xu, X. Yuan, and H. Qu, "Edge bundling in information visualization," *Tsinghua Sci. Technol.*, vol. 18, no. 2, pp. 145–156, Apr. 2013, ISSN: 1007-0214. DOI: 10.1109/TST.2013.6509098.

[218] M. Bilgic, L. Licamele, L. Getoor, and B. Shneiderman, "D-Dupe: An Interactive Tool for Entity Resolution in Social Networks," in *2006 IEEE Symposium On Visual Analytics Science And Technology*, Baltimore, MD, USA: IEEE, Oct. 2006, pp. 43–50, ISBN: 9781424405916,9781424405923. DOI: 10.1109/VAST.2006.261429.

[219] C. Dunne and B. Shneiderman, "Motif simplification: Improving network visualization readability with fan, connector, and clique glyphs," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '13, New York, NY, USA: Association for Computing Machinery,

Apr. 2013, pp. 3247–3256, ISBN: 9781450318990. DOI: 10.1145/2470654.2466444.

[220] C. Vehlow, T. Reinhardt, and D. Weiskopf, "Visualizing fuzzy overlapping communities in networks," en, *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 12, pp. 2486–2495, Dec. 2013, ISSN: 1077-2626,1941-0506. DOI: 10.1109/TVCG.2013.232.

[221] D. Liu, F. Guo, B. Deng, H. Qu, and Y. Wu, "egoComp: A node-link-based technique for visual comparison of ego-networks," *Inf. Vis.*, vol. 16, no. 3, pp. 179–189, Jul. 2017, ISSN: 1473-8716. DOI: 10.1177/1473871616667632.

[222] B. Kale, M. Sun, and M. E. Papka, "The state of the art in visualizing dynamic multivariate networks," en, *Comput. Graph. Forum*, vol. 42, no. 3, pp. 471–490, Jun. 2023, ISSN: 0167-7055,1467-8659. DOI: 10.1111/cgf.14856.

[223] Q. Liu, Y. Hu, L. Shi, X. Mu, Y. Zhang, and J. Tang, "EgoNetCloud: Event-based egocentric dynamic network visualization," in *2015 IEEE Conference on Visual Analytics Science and Technology (VAST)*, Chicago, IL, USA: IEEE, Oct. 2015, pp. 65–72, ISBN: 9781467397834. DOI: 10.1109/VAST.2015.7347632.

[224] Y. Wu, N. Pitipornvivat, J. Zhao, S. Yang, G. Huang, and H. Qu, "egoSlider: Visual Analysis of Egocentric Network Evolution," en, *IEEE Trans. Vis. Comput. Graph.*, vol. 22, no. 1, pp. 260–269, Jan. 2016, ISSN: 1077-2626,1941-0506. DOI: 10.1109/TVCG.2015.2468151.

[225] J. Portenoy, J. Hullman, and J. D. West, "Leveraging Citation Networks to Visualize Scholarly Influence Over Time," *Frontiers in Research Metrics and Analytics*, vol. 2, p. 8, 2017, ISSN: 2504-0537. DOI: 10.3389/frma.2017.00008.

[226] F. Reitz, *A Framework for an Ego-centered and Time-aware Visualization of Relations in Arbitrary Data Repositories*, Sep. 2010. arXiv: 1009.5183 [cs].

[227] K. Fu, T. Mao, Y. Wang, D. Lin, Y. Zhang, and X. Sun, "DyEgoVis: Visual Exploration of Dynamic Ego-Network Evolution," en, *NATO Adv. Sci. Inst. Ser. E Appl. Sci.*, vol. 11, no. 5, p. 2399, Mar. 2021, ISSN: 0168-132X. DOI: 10.3390/app11052399.

[228] L. Shi, C. Wang, Z. Wen, H. Qu, C. Lin, and Q. Liao, "1.5D Egocentric Dynamic Network Visualization," en, *IEEE Trans. Vis. Comput. Graph.*, vol. 21, no. 5, pp. 624–637, May 2015, ISSN: 1077-2626,1941-0506. DOI: 10.1109/TVCG.2014.2383380.

[229] G. J. Wills, "NicheWorks—Interactive visualization of very large graphs," *J. Comput. Graph. Stat.*, vol. 8, no. 2, pp. 190–212, Jun. 1999, ISSN: 1061-8600,1537-2715. DOI: 10.1080/10618600.1999.10474810.

[230] G. M. Draper, Y. Livnat, and R. F. Riesenfeld, "A survey of radial methods for information visualization," en, *IEEE Trans. Vis. Comput. Graph.*, vol. 15, no. 5, pp. 759–776, 2009, ISSN: 1077-2626,1941-0506. DOI: 10.1109/TVCG.2009.23.

[231] M. Xue, Y. Wang, C. Han, *et al.*, "Target Netgrams: An Annulus-Constrained Stress Model for Radial Graph Visualization," en, *IEEE Trans. Vis. Comput. Graph.*, vol. 29, no. 10, pp. 4256–4268, Oct. 2023, ISSN: 1077-2626,1941-0506. DOI: 10.1109/TVCG.2022.3187425.

[232] U. Brandes and C. Pich, "More Flexible Radial Layout," in *Graph Drawing*, ser. Lecture notes in computer science, D. Hutchison, T. Kanade, J. Kittler, *et al.*, Eds., vol. 5849, Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 107–118, ISBN: 9783642118043,9783642118050. DOI: 10.1007/978-3-642-11805-0_12.

[233] M. Farrugia, N. Hurley, and A. Quigley, "Exploring temporal ego networks using small multiples and tree-ring layouts," *Int Conf Adv Comput Interact*, pp. 79–88, Feb. 2011.

[234] T. Dang, P. Murray, and A. Forbes, "BioLinker: Bottom-up exploration of protein interaction networks," in *2017 IEEE Pacific Visualization Symposium (PacificVis)*, Apr. 2017, pp. 265–269. DOI: 10/gm5wzh.

[235] R. Yang, Y. Bai, Z. Qin, and T. Yu, "EgoNet: Identification of human disease ego-network modules," en, *BMC Genomics*, vol. 15, no. 1, p. 314, Apr. 2014, ISSN: 1471-2164. DOI: 10.1186/1471-2164-15-314.

[236] B. Lee, C. Plaisant, C. S. Parr, J.-D. Fekete, and N. Henry, "Task taxonomy for graph visualization," in *Proceedings of the 2006 AVI workshop on BEyond time and errors: novel*

evaluation methods for information visualization, ser. BELIV '06, New York, NY, USA: Association for Computing Machinery, May 2006, pp. 1–5, ISBN: 9781595935625. DOI: 10.1145/1168149.1168168.

[237] F. van Ham and A. Perer, "Search, show context, expand on demand: Supporting large graph exploration with degree-of-interest," en, *IEEE Trans. Vis. Comput. Graph.*, vol. 15, no. 6, pp. 953–960, 2009, ISSN: 1077-2626,1941-0506. DOI: 10.1109/TVCG.2009.108.

[238] B. Korte and J. Vygen, *Combinatorial Optimization: Theory and Algorithms* (Algorithms and Combinatorics), en. Berlin, Heidelberg: Springer Berlin Heidelberg, Jan. 2012, vol. 21, ISBN: 9783642244872,9783642244889. DOI: 10.1007/978-3-642-24488-9.

[239] M. Dörk, R. Comber, and M. Dade-Robertson, "Monadic exploration: Seeing the whole through its parts," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '14, New York, NY, USA: Association for Computing Machinery, Apr. 2014, pp. 1535–1544, ISBN: 9781450324731. DOI: 10.1145/2556288.2557083.

[240] D. Kato, *Jotai — Primitive and flexible state management for React*, https://jotai.org/, Accessed: 2024-2-8, 2023.

[241] *MUI: The React component library you always wanted*, https://mui.com/, Accessed: 2025-2-2, 2023.

[242] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using NetworkX," in *Proceedings of the 7th Python in Science Conference*, G. Varoquaux, T. Vaught, and J. Millman, Eds., Pasadena, CA USA, 2008, pp. 11–15.

[243] G. Hendricks, D. Tkaczyk, J. Lin, and P. Feeney, "Crossref: The sustainable source of community-owned scholarly metadata," en, *Quant. Sci. Stud.*, vol. 1, no. 1, pp. 414–427, Feb. 2020, ISSN: 2641-3337. DOI: 10.1162/qss_a_00022.

[244] N. T. Doncheva, J. H. Morris, J. Gorodkin, and L. J. Jensen, "Cytoscape StringApp: Network Analysis and Visualization of Proteomics Data," en, *J. Proteome Res.*, vol. 18, no. 2, pp. 623–632, Feb. 2019, ISSN: 1535-3893,1535-3907. DOI: 10.1021/acs.jproteome.8b00702.

[245] Z.-M. Xiao, D.-J. Lv, Y.-Z. Yu, *et al.*, "SMARCC1 Suppresses Tumor Progression by Inhibiting the PI3K/AKT Signaling Pathway in Prostate Cancer," en, *Front Cell Dev Biol*, vol. 9, p. 678 967, Jun. 2021, ISSN: 2296-634X. DOI: 10.3389/fcell.2021.678967.

[246] J. Ye, Y. Pang, X. Yang, *et al.*, "PPIH gene regulation system and its prognostic significance in hepatocellular carcinoma: A comprehensive analysis," en, *Aging*, vol. 15, no. 20, pp. 11 448–11 470, Oct. 2023, ISSN: 1945-4589. DOI: 10.18632/aging.205134.

[247] Z. Wang, H. Qiu, J. He, *et al.*, "The emerging roles of hnRNPK," en, *J. Cell. Physiol.*, vol. 235, no. 3, pp. 1995–2008, Mar. 2020, ISSN: 0021-9541,1097-4652. DOI: 10.1002/jcp.29186.

[248] Y. Xu, A. Nijhuis, and H. C. Keun, "RNA-binding motif protein 39 (RBM39): An emerging cancer target," en, *Br. J. Pharmacol.*, vol. 179, no. 12, pp. 2795–2812, Jun. 2022, ISSN: 0007-1188,1476-5381. DOI: 10.1111/bph.15331.

[249] M. C. Wahl, C. L. Will, and R. Lührmann, "The spliceosome: Design principles of a dynamic RNP machine," en, *Cell*, vol. 136, no. 4, pp. 701–718, Feb. 2009, ISSN: 0092-8674,1097-4172. DOI: 10.1016/j.cell.2009.02.009.

[250] A. Fabregat, S. Jupe, L. Matthews, *et al.*, "The Reactome Pathway Knowledgebase," en, *Nucleic Acids Res.*, vol. 46, no. D1, pp. D649–D655, Jan. 2018, ISSN: 0305-1048,1362-4962. DOI: 10.1093/nar/gkx1132.

[251] R. F. i. Cancho and R. V. Solé, "The small world of human language," *Proceedings of the Royal Society of London. Series B: Biological Sciences*, vol. 268, no. 1482, pp. 2261–2265, Nov. 2001, ISSN: 0962-8452,1471-2954. DOI: 10.1098/rspb.2001.1800.

[252] X. F. Wang and G. Chen, "Complex networks: Small-world, scale-free and beyond," *IEEE Circuits and Systems Magazine*, vol. 3, no. 1, pp. 6–20, 2003, ISSN: 1531-636X,1558-0830. DOI: 10.1109/MCAS.2003.1228503.

[253] V. Latora and M. Marchiori, "Efficient behavior of small-world networks," en, *Phys. Rev. Lett.*, vol. 87, no. 19,

p. 198 701, Nov. 2001, ISSN: 0031-9007,1079-7114. DOI: 10.1103/PhysRevLett.87.198701.

[254] P. Mahajan, O. Fiehn, and D. Barupal, "IDSL.GOA: Gene ontology analysis for interpreting metabolomic datasets," en, *Sci. Rep.*, vol. 14, no. 1, p. 1299, Jan. 2024, ISSN: 2045-2322. DOI: 10.1038/s41598-024-51992-x.

[255] P. A. Ewels, A. Peltzer, S. Fillinger, *et al.*, "The nf-core framework for community-curated bioinformatics pipelines," en, *Nat. Biotechnol.*, vol. 38, no. 3, pp. 276–278, Mar. 2020, ISSN: 1087-0156,1546-1696. DOI: 10.1038/s41587-020-0439-x.

[256] J. Ooge, G. Stiglic, and K. Verbert, "Explaining artificial intelligence with visual analytics in healthcare," en, *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 12, no. 1, Jan. 2022, ISSN: 1942-4787,1942-4795. DOI: 10.1002/widm.1427.

[257] B. C. Kwon, B. Eysenbach, J. Verma, *et al.*, "Clustervision: Visual supervision of unsupervised clustering," en, *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 1, pp. 142–151, Jan. 2018, ISSN: 1077-2626,1941-0506. DOI: 10.1109/TVCG.2017.2745085.