

Exploratory Visualizations of Multivariate Biomedical Data

Dissertation

der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

vorgelegt von
Nicolas Brich
aus Tübingen

Tübingen
2026

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

Tag der mündlichen Qualifikation:

30.07.2026

Dekan:

Prof. Dr. Thilo Stehle

1. Berichterstatter:

Prof. Dr. Michael Krone

2. Berichterstatter:

Prof. Dr. Sven Nahnsen

Abstract

This dissertation investigates the usage of *Exploratory Data Analysis* (EDA) in the biomedical domain. EDA was developed as a complementary approach to *Confirmatory Data Analysis* (CDA). While CDA mainly uses statistical analysis to confirm scientific hypotheses, EDA uses techniques like visualization to enable the exploration of data to guide hypothesis generation.

This is illustrated using three approaches developed and published for this dissertation. The first approach, squarely on the medical side of things, *ICUVA*, uses EDA to explore the status of the homeostatic system of a patient in an *Intensive Care Unit* (ICU) scenario. This status is described by multiple different types of ICU measurements. Usually, the course of these measurements across time is analyzed individually and often simple visualizations like line charts are used to explore the data. This is not feasible for more than a few measurements, without losing the completeness of the situation. Thus, *ICUA* uses time-curves, which embed the multi-variate data into a two-dimensional space to identify points in time when the system as changes as a whole.

To understand phenotypes such as disease phenotypes observed in clinical practice, the organism also needs to be analyzed on a molecular level. Various types of biomolecules are relevant for characterizing a phenotype. These molecules stem from different levels of gene-expression. Among those are transcripts, the messenger molecules copied from genes, proteins, the molecules that are synthesized from this information, and metabolites, small-molecules, like hormones, often produced with the help of enzymes. These molecules can be mapped to biological-pathways that describe how an organism works on a molecular level. In the second approach, *VisMOP*, such data of biomolecules mapped to biological-pathways is analyzed simultaneously by performing multi-omics clustering. These clustered pathways are visualized as a clustered, hierarchical graph where glyphs represent pathways and their multi-omics measurements. Ultimately, this helps to find similarly regulated

pathways which allows deriving hypotheses on how phenotypes are established.

The associations of proteins, as they are present in biological pathways can also be modeled as an association-network, commonly called *Protein-Protein Interaction* (PPI)-network. PPI-networks can for example be used to explore effects of defects in altered proteins on other proteins in the network, which can be used to generate hypotheses on disease mechanisms or other phenotypes. To facilitate such exploration, we developed *ProtEGOnist*, a domain agnostic approach that utilizes an ego-centric network layout to explore small world networks focusing on nodes-of-interest. We demonstrated this on PPI-network data.

The discussion shows how EDA applications can be used in a complex biomedical scenario, exemplified using the three presented approaches. Additionally, challenges when implementing EDA-applications in an academic settings are described. Finally, this dissertation addresses how we coped with these challenges in the presented approaches.

Zusammenfassung

Diese Dissertation erforscht die Nutzung der *Exploratory Data Analysis* (EDA) im Kontext biomedizinischer Daten. EDA wurde als komplementärer Ansatz zur *Confirmatory Data Analysis* (CDA) entwickelt. Während CDA hauptsächlich auf statistische Analysen setzt, um wissenschaftliche Hypothesen zu bestätigen, nutzt EDA Methoden wie Visualisierung um die Exploration von Daten zur Entwicklung neuer Hypothesen zu ermöglichen.

Dies wird in dieser Dissertation anhand dreier Ansätze aufgezeigt, die im Rahmen dieser Arbeit entwickelt und publiziert wurden. Der erste Ansatz, mit dem Namen *ICUVA* arbeitet auf Daten die deutlich in dem medizinischen Teil des Spektrums *biomedizinische Daten* zu verorten sind. Hier wird EDA genutzt um den Zustand des homöostatischen Systems eines Patienten im Kontext intensivmedizinischer Behandlung zu explorieren. Dieser Zustand wird durch eine Vielzahl an unterschiedlichen Messungen beschrieben. Normalerweise wird der zeitliche Verlauf dieser Messdaten durch individuelle, meist einfache Visualisierungen wie Liniendiagramme veranschaulicht und mit deren Hilfe exploriert. In diesem Ansatz ist es mit steigender Anzahl beobachteter Messereihen allerdings schwer das Gesamtbild nicht zu verlieren. Um dies zu ermöglichen, nutzt *ICUVA time-curves* in welchen die multivariaten Daten in einen zwei-dimensionalen Raum eingebettet werden, um so Zeitpunkte zu finden, an welchen sich das Gesamtsystem verändert.

Um Phänotypen, wie z.B. das Auftreten und die spezifischen Ausprägungen von Krankheiten, die in der klinischen Praxis beobachtet werden können zu verstehen, muss der Organismus auch auf einer molekularen Ebene analysiert werden. Hier sind unterschiedliche Arten von Biomolekülen zu berücksichtigen, die aus verschiedenen Phasen der Genexpression stammen. Dabei sind Transkripte, die Boten-Moleküle die von Genen transkribiert werden, Proteine, die aus diesen synthetisiert werden, als auch Metabolite, niedermolekulare Verbindungen, die häufig mithilfe von Enzymen produziert werden, zu nennen. Diese Typen von Biomolekülen lassen sich auf Stoffwechselwege, die

beschreiben wie ein Organismus auf molekularer Ebene funktioniert, abbilden. Der zweite Ansatz der in dieser Dissertation beschrieben wird, *VisMOP*, analysiert Messdaten dieser Biomoleküle im Kontext von Stoffwechselwegen mittels *multi-omics clustering*. *VisMOP* zeigt so geclusterte Stoffwechselwege als einen geclustertem hierarchischen Graphen in welchem Glyphen genutzt werden, um die enthaltenen Stoffwechselwege und die mit ihnen assoziierten multi-omics Messungen zu visualisieren. Dies hilft schlussendlich dabei Stoffwechselwege zu finden, welche ähnlich reguliert sind und damit Hypothesen aufzustellen, wie spezifische Phänotypen auf einer molekularen Ebene entstehen.

Die Assoziationen von Proteinen untereinander, wie sie auch in Stoffwechselwegen vorkommen, kann ebenfalls als Assoziations-Netzwerk modelliert werden. Diese Netzwerke werden als *Protein-Protein Interaction* (PPI) Netzwerk bezeichnet. Diese PPI Netzwerke werden z.B. genutzt um den Effekt defekter Proteine auf andere Proteine und deren Funktionen abzuschätzen und so Hypothesen zu Krankheits-Mechanismen und anderen Phänotypen zu erforschen. Um genau diese Funktionalität zu bieten wurde *ProtEGOnist* entwickelt. *ProtEGOnist* ist ein Domänen-agnostischer Ansatz, welcher Ego-zentrische Netzwerk-Layouts nutzt, um *small-world* Netzwerke ausgehend von einzelnen Startknoten zu explorieren. Im Kontext dieser Dissertation und der enthaltenen Publikation wurde dies auch anhand von PPI-Netzwerk Daten aufgezeigt.

Die Diskussion der Dissertation beschreibt, am Beispiel der drei entwickelten Anwendungen, wie EDA Anwendungen in einem komplexen Szenario verwendet werden können, die sich mit der Analyse von biomedizinischen Daten beschäftigen. Zusätzlich werden Herausforderungen diskutiert, die während der Entwicklung solcher EDA Anwendungen in einem universitären Rahmen auftreten können. Schlussendlich werden auch die Wege diskutiert, die genutzt wurden um mit diesen Herausforderungen während der Entwicklung der beschriebenen Anwendungen umzugehen.

Acknowledgments

I would first like to thank my primary supervisor Prof. Dr. Michael Krone and my secondary supervisor Prof. Dr. Kay Nieselt. Additionally, I would like to thank Prof. Dr. Sven Nahnsen for being an official reviewer of this thesis and for chairing the oral exam along with the other examiners of the oral exam.

Furthermore, I want to thank my scientific collaborators, that made contributions to all joint publications that make up this thesis and the other publications that were published during my work as scientific staff at the University of Tübingen.

Thanks also to my colleagues, Dr. Marco Schäfer at BDVA Group, and all colleagues from the Integrative Transcriptomics Group for all the daily lunches, the social gatherings, and emotional support during my PhD.

Lastly, I want to thank my family—my parents, my sister, and my partner—for supporting me in every endeavor from my childhood to now.

Contents

Abbreviations.....	VIII
1. Publications—Author Contributions.....	1
1.1 ICUVA.....	3
1.2 VisMOP.....	4
1.3 Protagonist.....	5
1.4 Other Publications.....	5
2. Introduction.....	7
2.1 Motivation.....	10
2.2 Contribution and Outline.....	12
3. Visualization—An Introduction.....	15
3.1 Why Visualize?.....	16
3.2 How to design Visualizations.....	19
3.3 Pitfalls of Visualization.....	27
3.4 Visualization and Web Technologies.....	28
4. MedVis—Visualization of Medical Data.....	31
4.1 Fundamentals of Medical Data.....	32
4.2 ICUVA – Visual Analytics of Multivariate Intensive Care Time Series Data.....	35
5. BioVis—Visualization of Biological Data.....	69
5.1 Fundamentals of Biological Data.....	70
5.2 visMOP – A Visual Analytics Approach for Multi-omics Pathways.....	87
5.3 ProtEGONist – Visual Analysis of Interactions in Small World Networks Using Ego graphs.....	113
5.4 Summary and Conclusion.....	143
6. Discussion.....	147
6.1 Lessons Learned.....	150
6.2 Conclusion.....	158
Bibliography.....	161

Abbreviations

CDA – Confirmatory Data Analysis II, IV, 8, 9

CVAE – Convolutional Variational Autoencoder 36, 44, 46, 53, 60, 61

DNA – deoxyribonucleic acid 70, 71, 72, 73, 74, 75, 76, 83

DOM – Document Object Model 29

DR – Dimensionality Reduction 37, 38, 39, 42, 43, 44, 46, 47, 50, 53, 58, 65, 67

ECG – Electrocardiogramm 32

EDA – Exploratory Data Analysis II, III, IV, V, 8, 9, 10, 11, 12, 13, 16, 17, 33, 143, 145, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158

EHR – Electronic Health Record 11, 13, 32, 33, 70, 148, 150

GC – Gas Chromatography 80, 81

HTML – Hypertext Markup Language 28

ICU – Intensive Care Unit II, 13, 33, 35, 36, 37, 39, 40, 41, 55, 58, 61, 66, 67, 148

JS – JavaScript 28, 29

LC – Liquid Chromatography 80

MDS – Multi-Dimensional Scaling 37, 38, 43, 44, 45

MS – Mass-Spectrometry 79, 80

NGS – Next Generation Sequencing 75

NMR – Nuclear Magnetic Resonance 80

PCA – Principal Component Analysis 37, 39, 43, 44, 45, 46, 53, 60, 61

PCR – polymerase chain reaction 73, 74

PPI – Protein-Protein Interaction III, V, 12, 14, 113, 115, 116, 117, 119, 120, 121, 131, 132, 135, 136, 140, 141, 143, 149, 152

RNA – ribonucleic acid 71, 72, 73, 74, 75, 76, 77, 83

RNA-seq – RNA sequencing 75, 76, 77

SMRT – Single-molecule real-time 76, 77

SNP – Single Nucleotide Polymorphism 71, 89

STAR – State of the Art Report 145

SVG – Scalable Vector Graphics 29

ToF – Time-of-Flight 79

UI – User Interface 29, 30, 156

UMAP – Uniform Manifold Approximation and Projection 38, 44, 46, 60, 61

UX – User Experience 29

VNA – Volume-Need-Analysis“ 40, 43, 56, 57

W3C – World Wide Web Consortium 28

cDNA – copy DNA 74, 75, 76, 77

mRNA – messenger RNA 72, 73, 74

miRNA – micro RNA 72

qPCR – realtime quantitative PCR 74, 75

rRNA – ribosomal RNA 72, 73, 75

tRNA – transfer RNA 72

1

PUBLICATIONS

Author Contributions



This dissertation is mainly based on four previously published works, one of which is an extension of another of these works, in which the author of this thesis was the primary contributor and, thus, the first author. The respective chapters are marked accordingly, and the following tables detail the respective author's contributions.

1.1 ICUVA

Section 4.2 uses the text and figures of the following publication. The author-contributions are as follows:

Author Position & Author	Scientific Ideas	Data Generation & Implementation	Analysis & Interpretation	Paper Writing
1. Brich, Nicolas	40 %	75 %	65 %	50 %
2. Schulz, Christoph	45 %	5 %	10 %	25 %
3. Peter, Jörg	0 %	5 %	5 %	0 %
4. Klingert, Wilfried	0 %	5 %	5 %	0 %
5. Schenk, Martin	5 %	5 %	5 %	0 %
6. Weiskopf, Daniel	5 %	0 %	0 %	5 %
7. Krone, Michael	5 %	5 %	10 %	20 %
Title of Paper:	Visual Analytics of Multivariate Intensive Care Time Series Data			
Publication Status:	published as: Brich, N., Schulz, C., Peter, J., Klingert, W., Schenk, M., Weiskopf, D., & Krone, M. (2020). Visual Analysis of Multivariate Intensive Care Surveillance Data. <i>Eurographics Workshop on Visual Computing for Biology and Medicine</i> . https://doi.org/10.2312/VCBM.20201174 and extended as Brich, N., Schulz, C., Peter, J., Klingert, W., Schenk, M., Weiskopf, D., & Krone, M. (2022). Visual Analytics of Multivariate Intensive Care Time Series Data. <i>Computer Graphics Forum</i> , 41(6), 273–286. https://doi.org/10.1111/cgf.14498			
A conditionally accepted version of the extended paper was also published as a chapter in the dissertation of Dr. Christoph Schulz [242].				

1.2 VisMOP

Section 5.2 uses the text and figures of the following publication The author-contributions are as follows:

Author Position & Author	Scien- tific Ideas	Data Gen- eration & Implemen- tation	Analysis & Interpreta- tion	Paper Writ- ing
1. Brich, Nicolas	50 %	70 %	55 %	60 %
2. Schacherer, Nadine	15 %	15 %	5 %	10 %
3. Hoene, Miriam	5 %	5 %	10 %	5 %
4. Weigert, Cora	5 %	5 %	5 %	0 %
5. Lehmann, Rainer	5 %	5 %	5 %	0 %
6. Krone, Michael	20 %	0 %	20 %	25 %
Title of Paper:	Visual Analytics of Multivariate Intensive Care Time Series Data			
Publication Status:	published as: Brich, N., Schacherer, N., Hoene, M., Weigert, C., Lehmann, R., & Krone, M. (2023). visMOP – A Visual Analytics Approach for Multi-omics Pathways. <i>Computer Graphics Forum</i> , 42(3), 259–270. https://doi.org/10.1111/cgf.14828			

1.3 Protegonist

Section 5.3 is largely based on the following publication The author contributions are as follows:

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 Using Ego-graphs			
Publication Status:	published as: Brich, N., Harbig, T., Paz, M. W., Nieselt, K., & Krone, M. (2024). ProtEGOnist: Visual Analysis of Interactions in Small World Networks Using Ego-graphs. <i>Computer Graphics Forum</i> , 43(3), e15078. https://doi.org/10.1111/cgf.15078			
This Paper was also published as a chapter in the dissertation of Dr. Theresa Harbig [107].				
* These authors contributed equally and should be considered as joint first authors.				

1.4 Other Publications

Besides the publications with a primary authorship, mentioned above, additional publications with co-authorships are referenced where fitting:

1.4.1 Research Papers

- Schäfer, M., Brich, N., Byška, J., Marques, S. M., Bednář, D., Thiel, P., Kozlíková, B., & Krone, M. (2023). InVADo: Interactive Visual Analysis

of Molecular Docking Data. *IEEE Transactions on Visualization and Computer Graphics*, 1–14. <https://doi.org/10.1109/TVCG.2023.3337642>

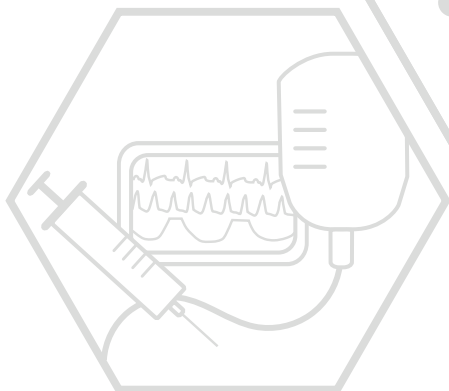
- In which the author contributed to the implementation of the web-frontend.
- Feyer, S. P., Pinaud, B., Kobourov, S., Brich, N., Krone, M., Kerren, A., Behrisch, M., Schreiber, F., & Klein, K. (2024). 2D, 2.5D, or 3D? An Exploratory Study on Multilayer Network Visualisations in Virtual Reality. *IEEE Transactions on Visualization and Computer Graphics*, 30(1), 469–479. <https://doi.org/10.1109/TVCG.2023.3327402>
 - In which the author contributed to the writing of the background section.

1.4.2 Reviews

- Scheer, J., Volkert, A., Brich, N., Weinert, L., Santhanam, N., Krone, M., Ganslandt, T., Boeker, M., & Nagel, T. (2022). Visualization Techniques of Time-Oriented Data for the Comparison of Single Patients With Multiple Patients or Cohorts: Scoping Review. *Journal of Medical Internet Research*, 24(10), e38041. <https://doi.org/10.2196/38041>
 - In which the author contributed to the review of sources, extraction and validation of data and visualizations as well as the writing of the manuscript, as described in the section “Authors’ Contributions” [239] .
- Ehlers, H., Brich, N., Krone, M., Nöllenburg, M., Yu, J., Natsukawa, H., Yuan, X., & Wu, H.-Y. (2025). An Introduction to and Survey of Biological Network Visualization. *Computers & Graphics*, 126, 104115. <https://doi.org/10.1016/j.cag.2024.104115>
 - In which the author contributed to the investigation and curation of sources, the original draft, the review and editing of the draft and the design of visualizations, as described in the section “CRediT authorship contribution statement” [70]

2

INTRODUCTION



“There is nothing better than a picture for making you think of questions you had forgotten to ask (even mentally)”

—John W Tukey and Paul A Tukey. [273]

Exploratory Data Analysis (EDA) has been a crucial step in scientific research for nearly 50 years [274]. EDA is the exploration of the data at hand using tools like visualizations to gather more insights into the data to form hypotheses. It was proposed as an additional step in experimental research, where *Confirmatory Data Analysis* (CDA), focused on statistical testing, is a commonly used approach [83]. In scientific research, many observed phenomena are yet poorly understood and the data is complex, including its generation and analysis. When trying to characterize these phenomena, the experiments oftentimes result in a vast amount of data generated without a clear and easily testable hypothesis.

As CDA emphasizes statistical hypothesis testing, the lack of a clear hypothesis often generates the need for an additional step beforehand. EDA can be used in such cases by providing a framework to generate hypotheses from complex or unknown data in the first place [83, 275]. And here in this process, as succinctly put by Tukey & Tukey [273], nothing is better than a visualization. Poorly understood or newly encountered phenomena occur in all domains of scientific research. In most domains, data is highly complex, in part fostered by the ever-increasing capacity to record and preserve large amounts of data. Biomedicine is a domain in which this is especially true [165]. Here, massive amounts of data are generated [205]. This is due to breakthroughs in data acquisition, like *high-throughput sequencing* combined with the increased capabilities of computational resources, as well as the ever-expanding boundaries of the research space [69]. Biomedical research often finds itself in a situation where an observed phenomenon and data gathered from it are

available but where hypotheses are hard to come by. This is due to the complex interplay of many biological systems and components making it difficult to postulate hypotheses that are supported by theory [188]. Some researchers then feel the need to use underdeveloped, and ill-fitting—with respect to theory—hypotheses to perform CDA. What they could do instead in such situations is to have *a look* into the data, performing EDA. Here visualizations can help using, among other things, the pattern recognition capabilities innate to us humans [289, 290], to develop well-founded hypotheses that they perform hypotheses driven experiments on.

Visualizations have been used to explore data from all domains, including biomedical data for at least 300 years [177, 191, 214, 218, 256]. But manually drawn visualizations, that are per definition static, require time and effort, thus limiting their use in exploratory analysis. However, the usage of computers and with them, interactive visualizations, increased the speed and ease with which new visualizations can be created, consequently amplifying their exploratory capacity. This development made visualizations a core tenet of EDA and a main driver for the exploration process [274]. In contrast to statistical tests, and CDA in general [83], most visualizations only have minimal requirements on the data. A statistical test requires a hypothesis and specific properties, such as specific knowledge about which groups are dependent or independent, which distribution underlies the data, or might be limited to a specific sample size [83]. For EDA, when such things are not known, it might be helpful to visualize the value distribution. There is a wide array of different established visualizations fit for value distributions. One such example is a histogram, for which a hypothesis is not needed, and the only meaningful requirement is that the data points are quantitative. This makes EDA an ideal first step in data analysis to generate a hypothesis to then apply CDA on a fresh dataset [83]. These lenient requirements also make visualization as a whole but also specific visualization techniques, like the just mentioned histogram, easily applicable to a wide range of scientific domains. The domains include, but are not limited to social

sciences [209], geography and demographics [8], as well as biomedical research [213].

This wide applicability and usefulness also lead to the development of the field of visualization as an independent research area. The development of an independent research area also leads to the formalization of many aspects. This includes how viewers perceive the world in the first place [289, 290]. Special focus is given on how to graphically design visualizations [271, 272]. Additionally, visualization research describes the tasks that a visualization should support to be of analytical merit [252] and how a user interacts with a visualization [251].

2.1 Motivation

In the biomedical domain researchers and other practitioners are often confronted with situations in which the data stems from poorly understood processes. Additionally, the analysis of such data is often a matter of life and death, which can not wait for the development of conventional analysis methods. Thus, biomedical data can be analyzed using EDA and, as an integral part of it, visualization. Furthermore, biomedical research and clinical practice go hand in hand in the current age of personalized medicine [237]. The data generated, both in experimental settings and in hospitals and medical practices, however, poses a challenge for data analysis both due to increasing amount of data and its increasing complexity [200, 205].

This is easily illustrated in the following highly idealized scenario. A patient gets admitted to the hospital in critical condition, the priority of the staff is to keep the patient alive. But while experience is valuable and used in clinical practice to administer critical care, many complex scenarios, like underlying rare or undiagnosed diseases, are still not caught and treated accordingly [27]. With the increasing amounts of data recorded bedside, EDA can be employed to thoroughly and quickly inspect the patient's vitals. Combined with additional analysis techniques and experience, this can lead to the discovery of the primary pathology. This direction, as a first step, is one that many techniques try to tackle.

Many visualization approaches in the context of EDA in the medical domain are concerned with the visualization of so-called *Electronic Health Records* (EHRs). EHRs are temporal, heterogenous—both in type and amount of data—digitally stored records of patient-care [54]. *LifeLines* [213] and its successor *EventFlow* [179], for example, proved to be very helpful tools for the exploratory analysis of EHRs. These approaches have proven themselves so helpful, that its application spreads beyond the scope of EHRs data analysis to topics only bordering biomedical data [61], as well as topics outside the realm of biomedical data, like the analysis motion capture data [24].

Usually only noticing the phenotypical abnormality in a patient's vitals is not sufficient to develop a treatment plan. In many cases, the pathology might be caused by genetic abnormalities or conditions, which interfere with the physiological metabolism of an individual. Personalized medicine, thus, researches the patient's genetic and molecular composition using omics measurement techniques, which measure the biological core molecules like DNA, proteins, and metabolites. These omics measurements generate tremendous amounts of data, data which due to the nature of the field, needs to be explored to generate a hypothesis for what is the underlying cause of the patient's disease. Many approaches already successfully apply the concept of exploratory visualization to the analysis of omics-data. One such approach can be seen in genomics explorers like *IGV* [227], to navigate the genome of the patient, or *Xena* [96], to inspect the genomes of cohorts of similar patients. They help navigate through complex high-volume genome-sequencing data to find and visualize, among other things, mutations in the genome, which can be causal for genetic disease or hallmarks of specific cancers.

While some of those methods utilize *de novo* data, most approaches rely on previous knowledge to relate the recorded data to. For this, years of biological research generated a vast amount of relational information about genes, proteins, metabolites, and their associated pathways, and the role they play in the context of disease. For example, a protein might be expressed in higher quantities than seen in the physiological, i.e., the healthy, state. When this protein, however, is not known to be related

to the observed pathology, a medical researcher can access databases linking proteins to known or theorized interaction partners with more documented information. One of the most widely used resources is the *STRING* [174] database, which uses a network with weighted edges to represent *Protein-Protein Interactions* (PPIs). Some of those databases, like *The Human Protein Atlas* [276] also offer additional information about the protein itself, like positioning in a metabolic pathway and information about the tissue in which the protein is mainly expressed. These databases contain large amounts of information in their own right, and EDA can make the discovery of such interaction partners more effective.

Even when these efforts are fruitful and the causal agent is found, it still might not be known well enough for a medication to be available that can effectively treat the disease. Analogous to biological databases, pharmacologists generate databases of millions of potentially pharmacologically active molecules. Algorithms and filters exist to whittle down the amount of potentially useful molecules for a given protein in a process called *hit-picking*. The results, still in the thousands, are oftentimes explored visually to generate a subset of laboratory-testable drug candidates [22]. Here EDA, shows the potential to be used to high effectiveness by allowing an exploratory workflow when performing *hit-picking*, both for the initial filtering as-well as the final visual exploration.

This scenario effectively illustrates that EDA, and with it exploratory data visualization, can be used at many entry points in the personalized medical pipeline. It must be noted, that each of the outlined steps is also an independent research direction in biomedical research and does benefit tremendously from EDA.

2.2 Contribution and Outline

This work compiles the investigation into the potential of explorative data visualization at multiple stages of biomedical research. This research process includes bedside data acquisition, i.e., phenotypical observations, multi-omic exploration of subject samples, exploring re-

lated but already compiled knowledge about biomolecules, and even the exploration of potential treatment options based on the interaction with specific, oftentimes misregulated, proteins. For this, several distinct exploratory visualization approaches and prototypes were developed.

This dissertation will first, in Chapter 3, examine the background of the main application domain, visualization research. The chapter details the history, design, and usage of visualization as a medium to convey scientific data while special emphasis is put on how visualization is used in the context of EDA.

With biomedical data-visualization, this work joins two related but distinct overarching topics: medicine and biology. Thus, the two domains and the developed approaches are described in two separate chapters (Chapter 4 & Chapter 5). These each contain the previously published work and required background to put the approaches into context.

The first of these chapters will deal with medicinal data in the form of EHR. For this, the approach *ICUVA* will be described (Section 4.2). *ICUVA* is a visualization approach assisting medical researchers and, potentially, clinicians to analyze complex patient states from highly multivariate *Intensive Care Unit* (ICU) data [36, 37]. At its core is the idea that a summarization of a set of vital parameters can be shown as a curve indicating the general status of a patient. Changes in this curve then indicate changes in the underlying status of the patient and might require action from the care personnel. It was tested with on EHR data acquired by a team of domain scientists in the experimental surgery department of the University of Tübingen.

The next stage in the outlined biomedical research pipeline—squarely in the domain of biological data—is presented in the subsequent chapter. This step is the inspection of the phenotype based on molecular biological methods. Here, multi-omics data, consisting of transcriptomic, proteomic, and metabolomic data act as the basis for the exploratory visualization. Utilizing this, we developed an approach allowing the integrative analysis of multiple omic measurements from experimental data in the context of metabolic pathways, *VisMOP* [35] (Section 5.2). As all biomolecule types in question can be mapped to those pathways,

we then generated an interactive visualization, in which different detail levels of metabolic pathways can be compared based on their biomolecule makeup.

Afterwards, we focussed on a specific type of biomolecule, as we designed a visualization approach aimed at investigating PPI networks from a target perspective [34] (Section 5.3). This means that instead of viewing the PPI network from a global perspective using general network metrics, we chose ego-centric exploration based on specific target proteins. During the development of this approach, we made sure that the resulting visualization offers general applicability to other domains featuring small-world networks, such as social or transport networks. These two approaches are then briefly discussed together.

Finally, Chapter 6 revisits the scenario presented in this introduction in context with the described approaches and discusses the broader influence of exploratory visualization on biomedical data analysis. This includes both insights gained from the approaches developed during this dissertation, and from other published work. Additionally, possible avenues for future research and development in the presented approaches are outlined.

3

VISUALIZATION

An Introduction



Visualization, as mentioned in Chapter 2, is the central technology that enabled EDA in the first place. And at its core, visualization is the non-written and non-verbal way of conveying information. Thus, it does not only apply to scientific data in the way it is discussed in the context of this dissertation, but to everyday life scenarios as well.

An intuitive example of information in need of visualization is given by *Tufte*—who taught alongside *Tukey* [272]—in his book *Envisioning Information* [271]. In a chapter aptly named *escaping flatland*, he describes how data visualization can be used with two-dimensional display-media, such as print or screen to communicate insights into multivariate data. He illustrates this, among other examples, with a table of acts of crime committed by prosecution witnesses in a mafia case. While normally the defense attorneys would need to list all those in a lengthy monologue, the joint display succinctly conveys the intended information: *the witnesses are criminals, and thus, not credible*. He achieves this is using the simplest form of data visualization, discounting the simple printing of values in a list, a table. Using more sophisticated visualizations would allow for a more nuanced way of conveying information.

However, abstracting multivariate data into two-dimensions is only one challenge in the process of generating a visualization to convey complex data. Owing to this, this chapter will discuss visualization as a discipline. This means revisiting why data is visualized and then detailing the principles and technologies that are used when designing and implementing visualizations. In that way, this chapter will act as a primer for visualization as a discipline. For this, this chapter will draw upon the seminal works of *Tufte* [270–272], *Munzner* [184] and *Ware* [289, 290]

3.1 Why Visualize?

While stated that the conveying of information is the main objective of visualization, the question remains *Why visualize?* Because, as described by *Munzner* [184], surely there are scenarios in which information can be conveyed clearly using text (or speech) alone. The result of a statistical test or from machine learning for example can be conveyed using a single

number or a simple sentence. Oftentimes, however, especially when there is no clear avenue to apply simple tests or algorithms it becomes imperative to visualize. This is also the case when an initial question concerning the data is missing, thus the importance of visualization for EDA.

But even if we already have such an initial question, visualization also can still be helpful, for example, by uncovering differences in seemingly similar data. Here many visualization researchers, including *Munzner* and *Tufte* show the poignant example of *Anscombe's quartet* [11]. Here *Anscombe* presents four two-dimensional datasets with identical descriptive statistics, means, variances, correlation and identical regression lines. The non-visualization, that is the “hard statistics” shows the identity between the datasets, by applying a simple visualization, a scatter plot, *Anscombe* shows the different nature of the datasets. These examples were extended later on to produce arbitrary shapes with a higher number of datapoints, for example one of a dinosaur [167]. While other statistical descriptors allow for disambiguation, like testing the dataset for fit to specific distributions, it is not always clear which descriptors would be adequate when encountering data for the first time, harking back to the initial problem of unclear analysis avenues.

In addition to the general question on “why visualize?” *Munzner* [184] also defines the role of the human and the computer in the loop of visualization, asking *Why visualize with a human/computer in the loop?* This is especially important for EDA as the interplay between computer and human in the visualization loop allows for interactive visualization. Here the computer and human can split analysis tasks, depending on their suitability for the strengths of the respective analyst. Humans can utilize their previous knowledge and their powerful pattern recognition and reasoning skills while offloading tedious or—with the scale of data in the “modern” age—impossible jobs like filtering or calculations to the computer. It then becomes imperative to briefly discuss the basics of perception, i.e., what makes humans good or bad in certain tasks.

3.1.1 Human Perception

Human perception encompasses multiple modalities, *vision*, *audition* and *tactition*, among ones that are less commonly used to describe data, like *olfaction* and *gustation*. It is quite obvious that the primary modality that is used for visualizations is human vision. Visualization can be combined with other modalities like acoustic signals, called *sonification* [76], or with *tactition* by using *haptics* or by actually generating an object via *physicalization* [124, 199, 201]. The other modalities like *olfaction* and *gustation* are harder to integrate with visualization, as among other things there is only limited and highly specialized hardware to convey them [258]. This is in contrast to haptic force-feedback hardware found in every smartphone or the ease of access to 3D-printers for *physicalization*.

A primary challenge in vision in general is the amount of visual data available at all times. This amount is large, in part, due to excellent visual perception capabilities of humans in many aspects, even compared to many other animals[46, 92]. Aspects of this are the high spatial resolution and the high amount of perceivable colors of human vision. And while humans have a rather narrow field of view, stereoscopic vision adds complexity through stereopsis, improving depth-perception [222].

To overcome this flood of information, Ware [289] argues that instead of retaining information of our surroundings in detail in our brain, humans evolved to rather *rescan* information when it becomes relevant for a given situation. But *rescanning* the overwhelming amount of information alone does not reduce the load on the brain. For this, Ware mentions a second mechanism: pattern recognition. By being good at analyzing patterns, humans are able to assess which part of the information presented, in this case the huge amount of visual stimuli, is relevant to a given situation.

Combining this pattern finding, with the *rescan* on-demand, humans thus only need to place comparatively little information into their *working memory* reducing the load on the brain allowing for the processing of complex scenarios. For vision this leads to the notion of *visual thinking*

which C. Ware [289] describes as “...a series of acts of attention, driving eye movements and tuning our pattern-finding circuits”.

By being aware of *visual thinking*, visualization designers can leverage the mechanisms of perception to create more effective visualizations. For an in-depth description see the book *Visual Thinking for Design* [289]. Munzner uses the concept of *visual thinking* in her book *Visualization Analysis and Design* [184] to provide a structured approach for designing visualizations, which will be described in the following sections.

3.2 How to design Visualizations

One way to design in a structured manner is the application of *typologies* and *taxonomies*. They can, for example, categorize the data, the purpose and the means of visualizations, to allow the (re-)application of established concepts. *Visualization Analysis and Design* [184] builds on such a typology introduced in the work of Brehmer and Munzner [33].

The typology is based on the three questions of *What?*, *Why?* and *How?* With the first of those questions they devise describe *What* kind of data is visualized. *Why* it is visualized, i.e., to which end. And of course *How* the visualization is built and interacted with.

3.2.1 What?

The first thing that is analyzed is the type of data present, the *What?* For this description Munzner introduces three concepts, *data*, *datasets* and *attributes* [184]. The data can take the form of several types like *items* or *links*. A biomedical example would be proteins as *items*, or interactions between those proteins as *links*. This type of data then influences the type of the dataset, like tables, when items, for example measured proteins, are simply listed, or networks, if the items are connected in a mathematical graph. Finally, oftentimes the items are associated with attributes, like protein mass in *dalton*. These attributes are themselves further categorized based on their type, which is *categorical* or *ordered*. Many attributes, often referred to as numerical data, are ordered and quantitative. Continuing with the biomedical examples, one

such attribute would be the molecular weight of proteins. Meaning that a 10 Da, 1 kDa and 1 MDa protein can be brought into an order and that the distance between them is proportional. Other attributes can be ordered but do not have a proportional distance between them. A Likert scale, common in usability-testing or surveys—*disagree*, *neutral*, *agree*—comes to mind. And finally unorderable data, for example classes of proteins *enzymes*, *Transcription Factor* and *Transmembrane Proteins*, that can be clearly differentiated, but that do not have any order associated with them.

To illustrate how the *What?* then influences the choice or design of a visualization we choose two of the previous examples concerning proteins. It would be very feasible to use a histogram to display the distribution of proteins, for example the total set of proteins expressed in an organism, by molecular weight. Trying to use the protein classes instead of molecular weight as a histogram would be impossible, as histograms require quantitative data. Yet, using a pie chart for the protein classes would be a better choice for categorical data.

3.2.2 Why?

The next question asked when analyzing or designing visualizations using the framework proposed by *Munzner* is the question, which tasks does—or should—the visualization support [184].

For this systematic ways of describing tasks are used. In visualization research there are a wide variety of task taxonomies and typologies of varying degrees of detail. The typology used by *Munzner* is a general purpose typology applicable to almost any visualization. In general this typology utilizes *Action-Target* pairs to split the tasks into. *Actions* describe what act is performed in the task, while *target* refers to a data characteristic to be acted-upon. In the previous example a possible *Action-Target* pair would be *Discover Extremes*, when having a look at a histogram of protein molecular weights.

This kind of very general and low-level task typology—the same holding true for taxonomies—often becomes less useful in domain-specific visualizations where the described tasks are commonly higher-level tasks.

There are for example task taxonomies for network visualizations [152], or even for biological pathways [185], and genomic visualizations [196]. Framing tasks and analyzing visualization in such a structured manner can help find aspects of a visualization, for example specific interactions, that make the required tasks easier to perform.

3.2.3 How?

As discussed here and in more detail by *Munzner*, the *What* and the *Why* fundamentally influence *How* a visualization is displayed [184]. In order to communicate how things are visualized, *Munzner* utilized several categories.

The most important one is the fundamental graphical primitives the *marks* [184]. Marks are zero-, one-, two- or three-dimensional objects like points, lines areas or volumes. Here it is imperative to use the correct mark for the data present. For example, a visualization designer should be careful not to use marks that visualize more attributes than the dataset actually contains. A common pitfall would be using an area mark where a point would suffice, as in area marks the area usually implies meaning. In order to style the marks, different *channels* are used. The used channels are fundamental aspects of the visualization like positioning or color of a mark, among others. Channels can be grouped in two categories, *magnitude* and *identity*, which should be used on ordered or categorical data respectively.

Channels are not created equal, meaning the members of both types of channels have different abilities to convey information. Here, two additional concepts are introduced by *Munzner*, the *expressiveness* and the *effectiveness* principles [184]. *Expressiveness* means that data must express what is present in the data and must not express what is not there. Here a straightforward example is the usage of a line graph for a set of categorical variables, where the line implies continuity between neighboring categories that is not there by definition. *Effectiveness* on the other hand means that the data variables, which should be emphasized the most, should be awarded the *best* channels.

But what means *best* in this case? *Munzner* ranks different channels by their effectiveness, i.e., how good information can be conveyed by a given channel [184]. The primary indicator, for magnitude channels, is the *accuracy*, i.e., how linearly changes in perception are tied to changes of the input stimuli. This ranking is influenced by psychophysics theory [261] and other empirical studies [52, 111].

For example, for ordered data, *Munzner* ranks positions on scales, as used in scatter plots, the highest, along with other common channels like length—think bar charts or box plots [184]. A lower-ranked channel is angle, for example found in pie charts. Other channels especially color, luminance and saturation, rank among the worst, paired with curvature and volume. These differences can be traced back to basic perception psychology. Here volume is a particularly low-ranking, as the human ability to gauge volumes, and differences in volumes is known to be rather poor, that is, non-linear [87].

Besides the accuracy, the three other aspects *Munzner* fields are: *separability*, *discriminability*, and the ability of a channel to cause *pop-out* [184]. *Separability* applies to both magnitude and identity channels and depends on the interaction of channels, i.e., how well can one read the information conveyed by a channel in the presence of another channel. For example, trying to encode values on both the *saturation* and the *luminance* will typically lead to interference, making it difficult to read any of the encoded values.

Discriminability is a self-interference measure, i.e., how easy it is to discern different marks using the same channel. That means poor discriminability requires more intense binning when using magnitude channels or, when using an identity channel, the need to reduce the amount of visualized categories. Here a well known-example is the usage of color hues encoding categories, where ideally less than 12 different hues should be used [109, 184]. Sometimes, however, it is possible to mitigate some discriminability problems by utilizing an additional channel. An example is the use of *jitter*, which is the introduction of positional variance in a previous unused direction. It is often used in categorical scatterplots (see Figure 1) to avoid overplotting in regions of high mark-density.

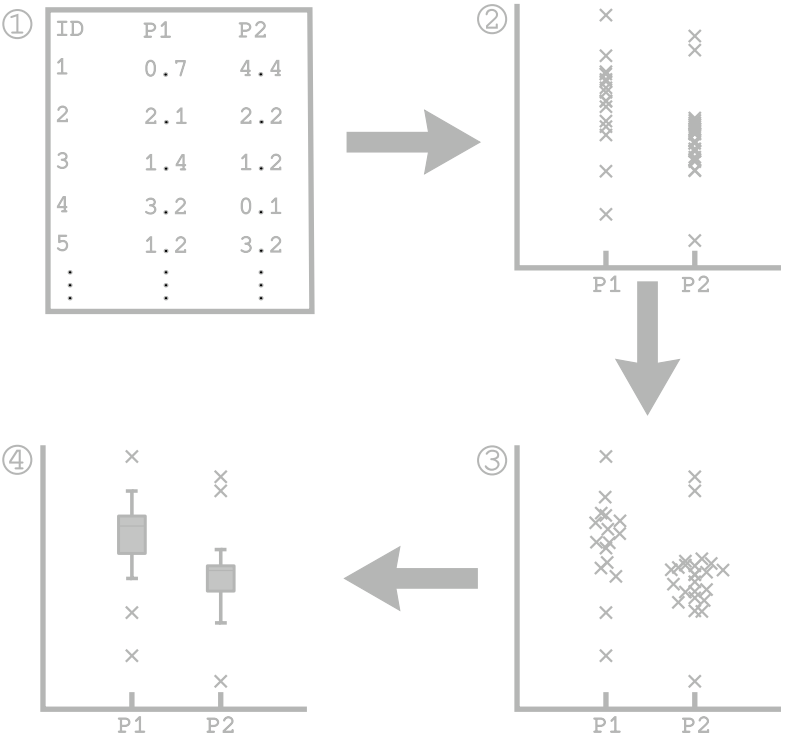


Figure 1: Encoding data ① with marks and channels reveals properties of the data ②. Using horizontal jitter, overplotting can be reduced to interpret the data ③. Using aggregation, data in dense regions can be summarized without losing much information ④.

Finally, the ability to *pop-out* can be an indicator of a channel's effectiveness [184]. This factor mainly applies to identity channels and can be considered an immediate and drastic form of discriminability. Here a channel is used to allow viewers of a visualization to quickly *select* a single item from a group of items. It should be noted, that similar to discriminability, the combination of channels can decrease the ability of a channel to cause the desired *pop-out*. These aspects are especially important when we consider that, depending on the type of visualization, the chosen marks are usually shown in large quantities.

Showing More

Sometimes, however, encoding data onto a single mark-channel combination is not representative enough of the process. That is, the data is too complex to encode it with a single type of mark without causing interference or occlusion. To alleviate this *Munzner* describes two basic approaches [184]. *Faceting*, which includes related techniques like superimposition, linked views and small-multiples and *Reducing*, which combines aggregation and filtering.

The basic goal of *reduction* [184] is to reduce the amount of elements shown as to not clutter the visualization, confusing the viewer. When talking *reducing*, the easiest approach is naturally to just not show specific data, for example by filtering. For that, however, the visualization designer needs to have some insight into what is shown or what regions of the visualization are of interest.

So to minimize data loss it might be a good idea to instead aggregate data [184]. Aggregating means, that a new datum is derived from multiple elements. A straightforward example is the box-plot. One can easily visualize several hundred data points in a scatter-plot, but this will cause considerable cluttering. By aggregating the data into a single glyph, that is a complex mark containing several basic marks, we can reduce this clutter while still retaining much of the original information. In this example (see Figure 1) by showing a box indicating quantiles, with whiskers indicating the distribution and single marks showing outliers.

The other way to reduce clutter, if the space needed for the visualization allows it, is *faceting* [184]. Here multiple visualizations are designed to visualize different aspects of the data simultaneously. With these, viewers of the visualization can utilize their pattern recognition to quickly switch between the visualizations depending on what aspect of the data they're interested in. Using the box-plot example, a designer could just as well generate one simple visualization for one aspect, e.g. the mean or the median of the items with a bar-chart, and another one, a histogram for each bar to describe the value distribution (see Figure 2).

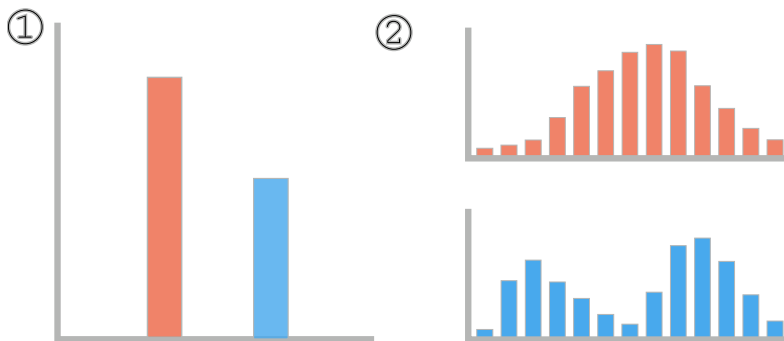


Figure 2: Showing the mean of two distributions as a bar-chart ①, while juxtaposing the values of those items as a histogram of the same distributions ②.

If a visualization is not cluttered superimposition can be used [184]. This means that another visualization—or combination of marks and channels—is overlaid on top of the first one. A straightforward example would be the superimposition of a regression line on top of a scatter-plot.

Interaction and Coordinated Views

Up until now the rules described mainly are concerned with what we would call *plots* that are—and were—commonly found in print or other static content. However, especially the techniques to accommodate more data, as described above, requires the designer to make considerable decisions concerning the message conveyed when applying them in a static context.

To shift this choice from the visualization designer to the viewer, which in this case would be the analyst in the scenario of exploratory data analysis, it is better to employ such visualizations in the context of interactive visualization systems. At the core of such systems is the interaction between the user and the visualization, which can be one or more coordinated views.

Let's consider the example from before (Figure 2). When using a bar-chart visualization, with a somewhat large number of bars, to display means of some dataset, showing a distribution for each bar requires either a lot of space, when faceting, or causes clutter, when superimpos-

ing the distributions on top of each other. If we rethink this example as an interactive visualization, it would be easy to just construct two linked views, faceted plots that react to changes caused by interaction. Here the bar-chart can act as a selector chart to switch between different histograms showing the distribution of the respective datasets. This principle can then be, of course, applied to many of the aspects discussed earlier, like filtering or aggregation. Even changing the fundamental building blocks of a visualization, the marks and channels, is a possibility to enhance the analysis capabilities.

The most powerful use of interaction is, however, when they are combined into multi-level exploratory visualization. Here, the *Visual Information-Seeking Mantra* described by Ben Shneiderman becomes very important. It states: “*Overview first, zoom and filter, details on demand*” [252].

But what does that mean from a visualization design perspective? It is applicable to very common scenarios, such as browsing and zooming a digital-interactive map, where integrating concepts of the *Visual Information-Seeking Mantra* [252] is straightforward, and the benefits are obvious. However, it is also very useful when being confronted with a scenario in which its integration is not as straightforward. One scenario is high-dimensional biological data, where it is not always clear what constitutes the *overview*, how to *zoom*, and which attributes should be used for the *details on demand*.

The first part, the overview, should contain all data but not in their raw form but rather in an aggregate or summarized fashion. This allows to detect a region of interest, which does not have to be an actual spatial region, in the larger dataset. In a next step, viewers of such a visualization will want to investigate this region of interest in more detail. Here *zoom and filter* makes its appearance. Zoom should not always be taken in a classical graphical sense, that is magnifying a specific part of the visualization to increase detail. Although, such a literal zoom is sometimes used, it is used more often in a metaphorical sense, for example by (partially) undoing aggregation operations. To avoid the inevitable visual clutter, less-interesting parts of the data can be filtered out. As a last step,

even more detail can be shown for the aggregated data. This can be the full extent of the original data point, summarizing statistics, related data points or the like. The type of displayed information is, just as the other two stages highly dependent on the domain and the aim of the analysis. From a practical point of view, there are several options to present such a multi-level visualization. In situations like the digital map mentioned above, using one view with interactive zooming to visualize the overview and the zoomed version can be useful to maintain the viewers' mental model, a space in which the viewer changes distance to the object of interest. In other cases, where the connection between the overview and the *zoomed-in* version are more abstract, it can be better to use faceted linked views. Similarly, the desired details can either be shown as additions to the zoomed in view, think hinting connections to other objects in a graph or as context menus and tooltips. Alternatively, they can be shown in a separate coordinated view, like a dynamic table.

3.3 Pitfalls of Visualization

Without discounting the reasons on why to visualize, one should not discount the possibility of visualizations failing and resulting in worse results than if other means of conveying information were used. For example, it has been shown recently [169], that visualization can cause a false sense of security when evaluating risk scenarios. Sometimes, visualizations are not created convincingly enough, which can sabotage important messaging and have real-life consequences. *Tufte* presents such a case, the infamous *Challenger* disaster, in which ineptly created visualizations hindered communications of research-engineers and decision makers, ultimately ending in tragedy [270].

Additionally, *Tufte* introduces two other concepts, one of the *data-to-ink ratio* and the related *chart junk* [272]. In essence by avoiding *ink*—in the current age more likely *non-background pixel color*—that is not based on data we, at the least, avoid bothering the viewer with non-data, and at best, avoid the viewer reading into this *chart junk*. *Chart junk* is the addition of elements to the visualization that do not add any

additional information [272]. An often seen example of chart junk is the (unjustified) usage of 3D elements in infographics [184]. At the forefront of this non-pattern is the infamous, 3D-piechart. Here the added volumes to the pie do not add any new information and even worse, causes people to perform bad when trying to decipher the shown proportions [253] —or as people are not particularly good at using pie charts in the first place [254], worse than bad.

A lot of pitfalls are covered by rules of thumbs, such as the ones presented by *Munzner*, stated by the visualization community over many years and projects [184]. Many of those pitfalls can be avoided by not violating rules outlined on *How to visualize*. By adhering to principles of good *expressiveness* and *effectiveness* visualization can avoid readers missing an important message because a subpar channel was used, or drawing unwarranted conclusions because of a channel mismatch to the data.

Of course, these “mistakes” can be performed deliberately to misguide viewers [272]. This *technique* even leaves open the avenue of *plausible deniability*. Thus, it is all the much more important for people to stick to *good visualization practice*, which is easier using (somewhat) formalized frameworks, to make such intentional failures more obvious.

3.4 Visualization and Web Technologies

While theoretical considerations are an important aspect of *how* to visualize, practical aspects are oftentimes equally important. The following section details why the works described in detail in this thesis, i.e., *ICUVA* (Section 4.2), *VisMOP* (Section 5.2) and *ProtEGOnist* (Section 5.3), are realized as interactive web visualizations.

Web technologies like *Hypertext Markup Language* (HTML) and *JavaScript* (JS), as well as other key standards defined by the *World Wide Web Consortium* (W3C), and the ubiquitous availability of those technologies on devices, allow interactive visualization to be available to large number of people, which made interactive visualization a mainstream phenomenon. In this regard, web technologies makes it easier to make research prototypes available along with their publication.

One example of this can be seen in the *TueVIS* server (<https://tuevis.cs.uni-tuebingen.de>), which hosts several research prototypes of multiple research groups at the *University of Tübingen*, like *ProtEGOnist*, which is part of this work.

An especially important development was the introduction of *Scalable Vector Graphics* (SVG). At their core, SVGs are instructions to draw shapes. Thus, their main benefit to raster graphics is, that they scale to the desired display resolution. And while this increases the fidelity of visualizations on the web it does not make its creation easier, thus not *per se* helping the dissemination of visualization to the masses. What does make the creation of visualizations easier, the encapsulation of common marks into higher-level instructions. So instead of drawing a rectangle as a series of four lines, the higher-level instruction of a *rect* can directly be drawn with a position and vertical and horizontal dimensions.

Combining this way of rendering items and the programmatic access to the *Document Object Model* (DOM) and with it parameters of an SVG allows programmers to generate SVG elements from data. This fact spawned the development of the JS library *d3.js* [29]. *d3.js* used the concept of encapsulation in higher-level by supplying functions directly binding data to DOM-elements. This allows for interactive visualization by providing transitions when the data changes. Additionally, *d3.js* also supplies a wide variety of even higher-level instructions for shapes or axes and even layout algorithms among many other things.

And while rudimentary control elements can be created using *d3.js* it was not explicitly designed for the generation of *User Interface* (UI)-elements. However, in order to supply interactive and ideally exploratory potent visualizations, it is imperative that they are designed in a way such that users can easily interact with it, i.e., good *UI/User Experience* (UX)-design. And as UI-design is a whole discipline in of itself it is not really feasible for visualizers to design each UI element themselves. Thus, the obvious thing is to draw upon tested components, similar to delegating rendering to the browser via SVG.

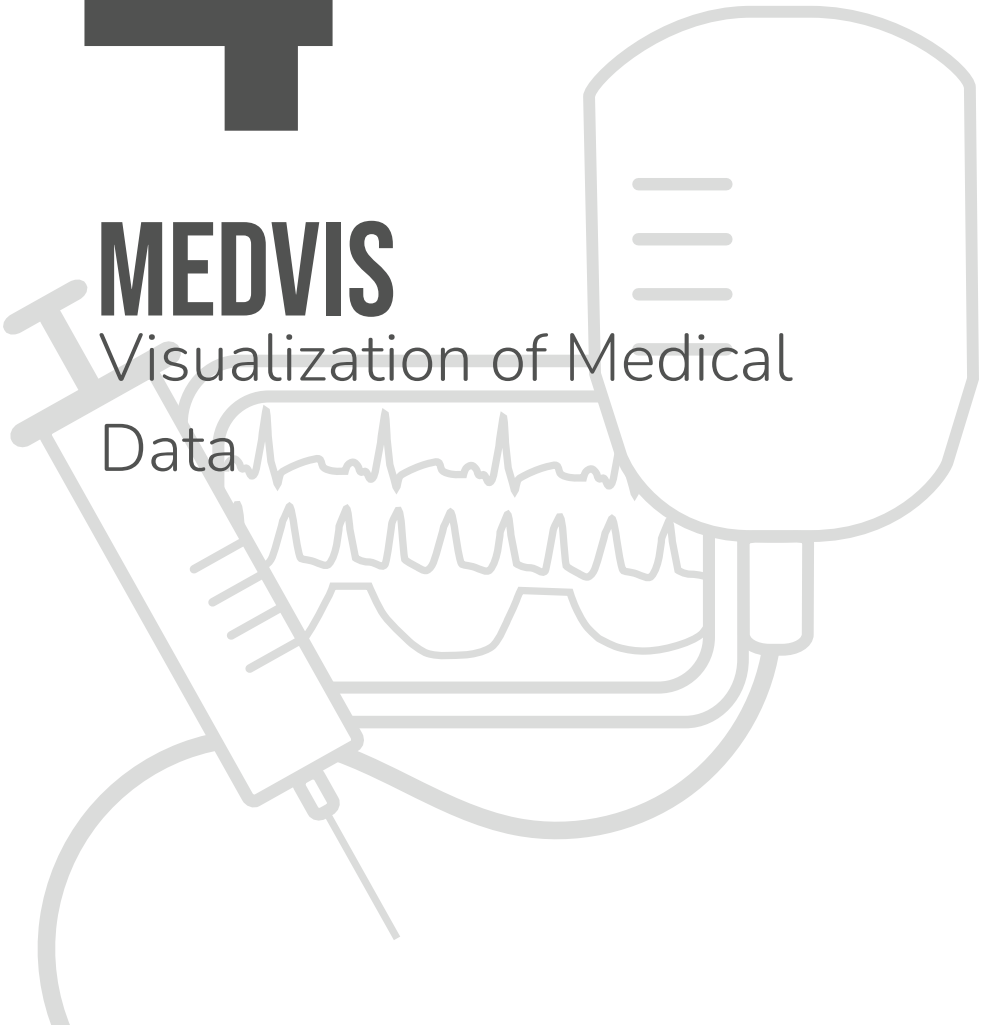
Such tested components are found in so-called front-end frameworks. Among the most tried of these frameworks are *Vue* [1] and *React* [78],

which were used in this work, both featuring large ecosystems. Integration with those frameworks goes far beyond simple UI design and indeed concerns with almost all aspects of the visualization design for web-browsers.

4

MEDVIS

Visualization of Medical
Data



4.1 Fundamentals of Medical Data

Medical data is data that describes patients or population health. As there are many facets of what constitutes health and just as many methods to assay it the resulting data are very heterogeneous. For example blood might be analyzed to check for nutrient deficiencies or ultrasound may be performed to check for gallstones, resulting in numerical or image data, respectively.

This data is often compiled digitally in *Electronic Health Records* (EHRs) to give a comprehensive insight into a patients state. Amongst others, EHR includes two important subgroups of longitudinal data: Event-type data and vital-type data [285]. An example for event-type data is a visit to a doctor's office with a diagnosis—an event that then can be recorded. This doctor might then refer the patient to a specialist, like a cardiologist, for further treatment. Consequently, the specialist visit is also recorded.

During the cardiologist appointment vital-type data, for example patients *Electrocardiogramm* (ECG) might be recorded. An ECG records the currents used to innervate the heart—in its function as a *muscle pump* [45]—and thus can be used to draw conclusions about coronary health. This, longitudinal, vital-type data is added to the EHR to give quantitative information about the patient. Additional examples of this vital-type data are common blood tests to determine inflammation values or signs of infection, or basic parameters like lung function, heart rate, and blood pressure.

It is apparent that this type of vital-type data varies in complexity. Some parameters can be understood and analyzed without any further processing like heart rate and blood pressure. Other types of vital-type data, however, need either very specific equipment or substantial processing of the data to be of any analytical value. This includes things like the already mentioned blood test values, which are not performed by laypersons or even doctors but by laboratory personnel. To make sense of the data recorded by the laboratory personnel, doctors then need to interpret this data to gauge the relevance and impact on the patients' health.

4

An additional aspect that has to be considered when interpreting medical data is the quality of the data. While the data in research settings is oftentimes of high quality, this cannot be said from day-to-day clinical work. One aspect of data-quality is the amount and completeness of data recorded. For example, in an experimental medical setup, the experimental design requires the data to be as complete and consistent as possible, often requiring replicates to allow for statistical evaluation. While in clinical reality data amount and data completeness collected in an ICU scenario are of little concern, as the main objective is ensuring the patient's wellbeing via *ad-hoc*-monitoring and not for *post-hoc*-analysis to further medical research. Also, experimental procedures are oftentimes performed on animal surrogate models [141], which do not need the same adjuvant treatment, altering the recording of the vitals, which would be the reality of actual, real-life, clinical data recording. So any *post-hoc*-analysis approach, like an EDA approach, would need to accommodate for different realities concerning data availability and quality. Additionally, data is not recorded with the same frequency, e.g., heart rate is recorded continuously while blood-gas-analysis is only performed every few hours, at best. This makes many analysis techniques more difficult, as not every point in time has the same amounts of data available.

In general, solutions exist to address the data quality and availability issues [23]. The techniques, however, must also be appropriate for the medical-field, where wrong assumptions can have negative consequences. For example, in many sciences, interpolation is used to fill short holes in the data or predict trends if data is only recorded for a smaller segment. Interpolation, however, always uses assumptions to create data that was not actually recorded. Thus, in the medical field it can be better to utilize simple methods like the last recorded value, or using the closest-value, or not using interpolation at all while being aware that this introduces some degree of bias [21, 23].

While medical research can have multiple objectives, oftentimes correlating patient outcome with EHR and/or vital-type data is at least part of the story. For example, both types of data can be utilized to compare

the patients with each other or groups of patients to characterize a patient's condition. This was the focus of a scoping review performed in conjunction the Hochschule Mannheim [239]. Even though we primarily focused on vital type-data in the ICUVA publications [36, 37], there were a host of synergistic effects concerning knowledge acquired during the development of ICUVA as well as new perspectives gained from the scoping review for future work in the ICUVA-project.

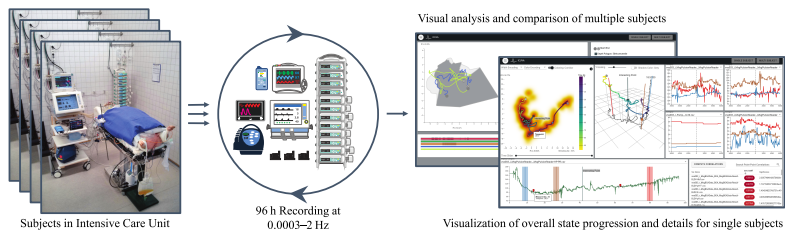


Figure 3: Our visual analytics application for multivariate medical data applied to data obtained during an experiment with porcine subjects attached to multiple medical devices in an intensive care unit. The overall state progression over the set of measurements per animal is shown as time-curve-inspired plots. A line plot showing a selected measurement and small multiples composed of selection-dependent line plots allow for state comparison and in-depth analysis. For an ensemble of multiple subjects, a time curve boxplot allows to compare subjects.

4.2 ICUVA – Visual Analytics of Multivariate Intensive Care Time Series Data

This Section contains previously published text

The following text, figures, and tables were published as [36] N. Brich, C. Schulz, J. Peter, W. Klingert, M. Schenk, D. Weiskopf, and M. Krone. 2022. Visual Analytics of Multivariate Intensive Care Time Series Data. *Computer Graphics Forum* 41, 6 (September 2022), 273–286. <https://doi.org/10.1111/cgf.14498>. Minor changes were made. These are the removal of the abstract, the funding information as well as the inclusion of the material previously placed in the supplementary material.

In a modern clinical setting, patients in an ICU are closely monitored by multiple medical devices. This multitude of measurements has to be communicated for interpretation, either to medical personnel for progressive analysis (monitoring), or—as in our context—to medical researchers for post-hoc analysis (experiments). Humans have difficulties aggregating more than three quantities, especially under the presence of uncertainty [99]. However, multiple parameters are often needed to provide an overall impression of the condition of a patient. Only consid-

ering a combination of changes in multiple parameters will hint toward important transitions of the whole homeostatic system. For instance, individual parameters might seem uncritical if considered independently, but in conjunction, can indicate an imminent circulatory shock. Ideally, an illustration of the patient state would be based on the entirety of time-varying interacting variables, without having to rely on a set of specific, pre-defined parameters. Moreover, such an approach would reveal high-dimensional processes clearly without being overly specific and still provide fine-grained details on demand. In practice, one of the primary problems with multivariate time-varying data is the conceptual and physical limitation of straightforward visualization techniques: most established approaches, such as scatter plot matrices, do not scale well enough to support the assessment of more than a few variables due to visual clutter and required mental effort to draw the right conclusions. In response to this observation, many expert users resort to univariate visualization techniques and (over-)simplifying derived characteristic quantities that take little advantage of the human visual system. In this work, which is an extension of an earlier workshop paper [37], we propose a visualization approach that facilitates the exploration and analysis of multi-dimensional, time-dependent ICU data based on time curves [15, 74], small multiples [272], and comparative views for an ensemble of subjects. We showcase our approach using real-world data obtained from an experimental surgery study performed on a porcine surrogate model (Figure 3). To compute our time curves, we have analyzed various projection techniques regarding their suitability for our application. Additionally, an unsupervised machine learning model—a *Convolutional Variational Autoencoder* (CVAE)—was implemented to better account for non-linearity and uncertainty in time curves. To disambiguate self-shadowing and resolve occlusion of the time curve, we implemented a linked 3D visualization in which the time curve is visualized in a space-time cube. Different coloring modes for the time curve facilitate tracking single variables and connecting trends in the overall patient status with these variables. To compare patient states and drill down into details, our application allows the user to select multiple data points in time that, together with a user-defined temporal context, are then shown

using small multiples. To quantify the similarity of changes of a specific variable at different time points, the point-to-point Pearson correlation can be shown. The contributions within our web-based visualization application can be summarized as follows: the visual analysis approach to high-dimensional and time-varying ICU data, the adaption of various visualization techniques using z-score and data-depth-based statistics to ease outlier analysis, a domain-driven generative data model to evaluate the applicability of various *Dimensionality Reduction* (DR) algorithms to ICU measurements, and a discussion containing expert feedback as well as an outlook for clinical surveillance.

4.2.1 Related Work

Although the visualization of high-dimensional and time-varying data has been investigated intensively and has been reviewed in many extensive surveys [4, 136, 157], it is still an active area of research. Classical multivariate visualizations such as parallel coordinate plots and scatter plot matrices are either a poor fit for visualizing multivariate time series, or only suitable for specific tasks like finding correlations between single variables. Thus, tailored task-driven visualization methods are needed [184] to facilitate an effective visual analysis of complex multivariate time-varying data. Since we are dealing with a number of time-varying input dimensions exceeding the capabilities of classical multivariate visualization, we briefly review suitable approaches.

Dimensionality Reduction

DR methods compute a low-dimensional representation from high-dimensional data, which are often easier to visualize and interpret than high-dimensional data. A typical example of this is *Principal Component Analysis* (PCA)[120, 206], which computes a linear projection based on variance. Another widespread technique is *Multi-Dimensional Scaling* (MDS)[28], which optimizes for representing high-dimensional distances in a low-dimensional space. Generally, there is neither a perfect solution nor a uniquely defined global optimum for globally preserving distances. Thus, many techniques that use neighborhood information instead of

distance have been developed. A popular example is the computationally expensive t-Distributed Stochastic neighbor Embedding (t-SNE) [164], which optimizes using entropy, generally leading to a misrepresentation of global distances in the embedding. However, t-SNE and other non-linear techniques excel at separating clusters. A more recent approach is *Uniform Manifold Approximation and Projection* (UMAP)[172], which aims at reducing this misrepresentation of distant data points by using a Riemann manifold. Vernier et al. [281] investigated a wide variety of drs regarding their feasibility towards multivariate temporal data. Our application guides the data-driven choice of DR technique, based on approaches from research on quantification and visualization of projection error [114, 245, 286]. To gain a better understanding of the overall suitability of the DR techniques for our application, we use a domain-specific generative data model for evaluation purposes [241] (Subsection 4.2.4, Subsubsection Dimensionality Reduction). Vernier et al. found evidence that variational autoencoders [139] are a suitable choice for DR of multivariate temporal data. Since it depends on the data and the research question which of these approaches is most suitable, our application offers the choice between linear (focusing on temporal change) and non-linear techniques (focusing on clusters).

Visualization of Multivariate Time Series

Our medical visual analysis application was inspired by several techniques to visualize multivariate time series that leverage DR: Jaeckle et al. [122] use temporal MDS to project non-temporal dimensions into a single dimension while retaining one display dimension for the mapping of the time component. This results in an optimal depiction of the progressing time, but impedes the distinction of different states, since they are plotted on only one axis. As we want to provide an intuitive overview of differences in patient states, the utilization of the two major axes for non-temporal components seems like a better choice. Bach et al. [15] concurred with this assessment and developed the time curve concept. They used metric MDS to project the high-dimensional data in two dimensions while illustrating the temporal succession by connecting consecutive points using a Bézier curve. A similar approach was previ-

ously presented by Bernard et al. [25], who used PCA. Concurrently to Bach et al., van den Elzen et al. [74] developed a similar technique to visualize the temporal evolution of graphs. These techniques can be classified as connected scatter plots, which follow the same idea of connecting temporally consecutive data points, but do not use DR and date back to 1790 [108]. An example is the work of S. Grottel, J. Heinrich, D. Weiskopf, and S. Gumhold. [100], who map time to line density in traditional scatter plots and parallel coordinates. Recently, time curves have been used by Hinterreiter et al. [117] to analyze solutions in problem-solving tasks. We adopt the concept of time curves and integrate it into our exploratory data analysis application. Due to the oscillating nature of the ICU data, we extend the original concept of time curves with filtering. We use this filtering step to declutter the data—and, consequently, the resulting visualization—and to remove outliers. Our application can handle not only single data sets but also multiple ones to compare the ensemble members. For the analysis of multiple data sets, we extend the time curve visualization with the notion of data depth and curve boxplots [178]. Data depth acts as a measurement to describe how central a single data point is within the whole data set: the higher a data depth, the more central a given data point is [58]. While data depth was already successfully applied to time series [162, 163], we explore the application of this concept to time curves.

Medical visualization

Ordonez et al. [198] explored the usage of star charts for visualizing ICU data. While this works reasonably well for a small number of variables, visual scalability is limited. In the course of the increasing digitalization, the visualization of electronic health records (EHR) has gained much attention. Zhang et al. [304] proposed a visual analytics framework that unifies EHR information for use in emergency rooms. While it also shows time-varying multidimensional data, their approach is not scalable and assumes discrete patient states and transitions, not numerical measurements. Similarly, Rind et al. [225] presented a tool for the visual analysis of long-term medical records of patients with chronic diseases. Their application case is different from ours since the data is more sparsely

sampled over much longer periods of time. Ten Caat et al. [42] introduced the concept of tiled parallel coordinates for multichannel EEG data. While their data is more similar to ours, they just have multiple sensors of the same type that all measure at the same frequency. They also use aggregation over the whole measurement time, which is not suitable in our case.

4.2.2 Medical Background

In this section, we summarize the experiment we used to evaluate our approach and the required clinical background knowledge [141, 211, 212]. Preliminary research showed insufficient knowledge regarding the assessment of the volume state in clinical practice for hospitalized patients. The *Volume-Need-Analysis*“ (VNA) study was conducted to answer research questions regarding the reliable assessment of volume states in a clinical context. One such question is whether it is possible to predict states and potential problems by observing measurable quantities of the organism so that doctors can proactively intervene, instead of being forced to react. To this end, a set of ICU devices commonly found in a clinical context was used to obtain the measurements. However, the setup was far more comprehensive than a regular setup in a hospital, providing almost complete knowledge about the volume state of subjects. The VNA study was conducted using 10 fully anaesthetized pig surrogates in accordance with ethical guidelines. In total, the experimental setup measured roughly 250 parameters (all of which are *vital-type* data as described in Section 4.1) over roughly 4 days each, with two experiments being ended prematurely. Recorded parameters include a variety of cardiovascular parameters such as blood pressure and heart rate, blood gas analysis results such as ion and carbohydrate concentrations, as well as general parameters like weight and diuresis volume. Initially, the animals were given defined amounts of fluid using intravenous infusions to reach an increased fluid state. Then, fluid supply was stopped, and the body fluid was reduced through diuretics. The fluid state was assessed at defined states by performing a variety of tests. All measurements were collected from a set of nine devices via manufacturer interfaces or wiretapping. While some devices sample

adaptively, the average sampling rate varies between $1/3600$ Hz (blood gas analysis) and 1-2 Hz (bedside monitors). All devices are synchronized to an accuracy of approximately one second.

4.2.3 Requirements

The measurement data already defined the quantity structure (see Subsection 4.2.2) therefore, we can focus the requirement analysis on the functionality. The high-level goal of our collaboration is to provide *an orthogonal approach to visual analysis of individual measurements*. In line with our experts, we argue that a loss of information to gain insights into high-dimensional processes is justifiable: as our primary use case is the post-hoc analysis of ICU data, our system cannot cause harm to the subjects. Thus, the following goals and requirements were defined iteratively during multiple design sessions and on-site meetings together with our medical domain experts:

- R1** The main objective is to be able to tell whether the patient is in a stable state or not. Since a patient does not necessarily have to be within the ideal parameter ranges, communication of change rates and state comparison is more important than absolute values.
- R2** Both the aggregate of change and individual changes can be of interest. Therefore, the visualization application should allow for both a top-down and a bottom-up exploration of the data.
- R3** Robust statistical methods should be used to guide the search for interesting patterns and relevant behavior—some parameters are heavy-tailed distributed and oscillating.
- R4** Medical researchers have high expectations regarding sanity-checking and explanation of phenomena due to the nature of the medical domain—easy interpretation without a strong mathematical background is of major interest.
- R5** It should be possible to put individual patients in the context of an ensemble to assess if a patient is representative or exceptional.

4.2.4 Visualization of ICU Data

Our goal was to combine several visualization approaches in one application tailored to medical experts. In this section, we describe the techniques we use as components and their adaption to our specific application. In accordance with the visualization pipeline [181], this includes data cleansing, an evaluation of DR algorithms, and a description of the visual mappings, which consist of filtered time curves and small multiples composed of line plots for in-depth analysis and comparison of patient states, as well as time curve boxplots for ensemble analysis (Figure 3).

Data Cleansing and Preprocessing

We cleanse the data beforehand to remove measurement errors. We remove erroneous samples using a list of user-defined error codes, filter inaccurate samples, and remove categorical time series and those that have a constant value. To identify and remove outliers, we have explored various techniques, such as the Savitzky-Golay filter [238] that approximates a variable using a polynomial. While this filter removes severe outliers, it also introduces artificial values and thus conflicts with **R4**. We opted for a z-score-based filter that classifies values as outliers that are not contained in the hull spanned by the moving average and standard deviation. All considered DR techniques require a uniform sampling, i.e., all components of a high-dimensional point have to be defined. Due to the high variance in sampling rates, a simple downsampling approach would lead to massive loss of data. In line with recommendations from our medical experts, we do not use interpolation to fill in missing values since this would introduce a possibly invalid assumption about the measured quantities. Therefore, we unify sampling rates across all time series. Given a user-specified sampling rate, we down-sample time series with a higher sampling rate than the chosen one and up-sample time series with a lower sampling rate, respectively. Missing values are filled via forward-filling (i.e. the last valid value is used until the next valid value is reached). Our medical experts approved of this strategy. If

a measurement starts at a later point in time, we apply backward-filling, as there are no earlier samples available.

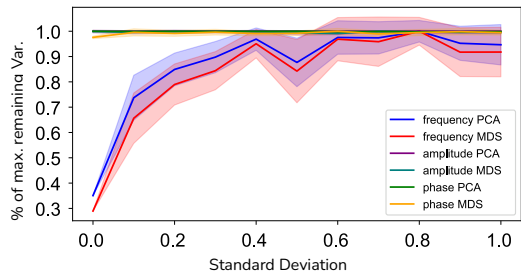
Dimensionality Reduction

Time curves project a time-varying high-dimensional space into a two-dimensional space, where time is not part of the DR, i.e., $(t, \mathbb{R}^n) \mapsto (t, \mathbb{R}^2)$. Time is added again after the DR by connecting the points in chronological order using lines. Therefore, we had to choose a suitable DR technique. To evaluate the DR, we use a generative data model [241], since the data from the VNA experiment contains unknown trends and patterns. Our data generator emulates the properties of the expected real-world multivariate time series using a combination of sinusoidal functions and piecewise linear ones. To justify the validity of our generator, we refer to the parameter range of the devices used during the experiment, as well as expert feedback. Thus, the resulting time series are representative of experimental measurements, as they contain variables with repeating patterns of varying frequency and amplitude (e.g., pulse, respiration) in conjunction with aperiodic, linearly changing quantities (e.g., temperature, weight). This approach allows us to establish a link between the parameters of the generative model and the output of the DR algorithm. This link is established through an evaluation metric: for PCA, it is the length of the first two components; for MDS, it is stress. For non-linear techniques, interpretation of the resulting dimensions is inappropriate. This form of sensitivity analysis would not be possible using only a few data sets with unknown properties. Moreover, the generative data model helps us to do sanity checking and allows users to familiarize themselves with the application using easy-to-understand data (**R4**). For evaluation, we generated six synthetic time series. To model physiological processes, which often exhibit oscillating behavior, the synthetic data sets are sinusoidal curve ensembles with varying frequency, amplitude and phase. Generally, we found that MDS using Euclidean distances represents long distances better than short ones, causing the beginning and end to be slightly deformed. Note that we used metric MDS with the SMACOF algorithm [153], which—in contrast

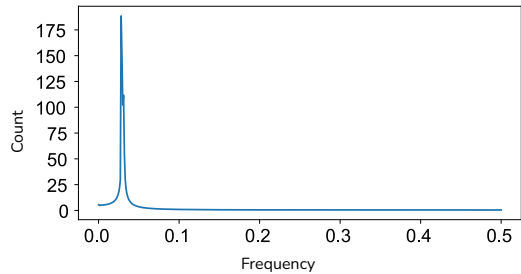
to classical MDS [267]—differs from PCA. Moreover, MDS tends to introduce loops that are not present in the data, which might be caused by non-optimal flipping and rotation while iterating (unfavorable local optima). PCA, on the other hand, shows the oscillating behavior of the data while separating dissimilar clusters of time series. We investigated the influence of varying frequency, amplitude, and phase (model parameters). We found that amplitude and phase had little impact on projection quality, while varying frequencies quickly degraded projection quality (Figure 4).

While both metric MDS and PCA are linear techniques, we suspect that suboptimal preconditioning leads to favoring PCA. Considering both aspects, we rate PCA as the suitable default and more defensive choice for DR in our application (**R3**). This is in line with van den Elzen et al. [74], who used PCA to facilitate interpretation of the resulting dimensions, and t-SNE for the analysis of clusters. When a more cluster-oriented approach is desired, we offer UMAP as a non-linear DR method. It yields comparable results to t-SNE while exhibiting a more desirable performance profile.

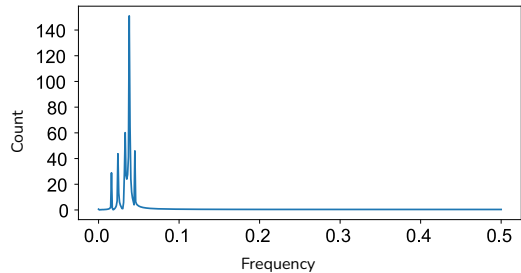
CVAEs are an unsupervised machine learning method utilizing an encoder-decoder architecture, each consisting of several layers, and a variational constraint connecting the encoder and decoder. Our data preparation and layer design is inspired by Ali et al. [5] and the Keras documentation [47]. Similarly to Ali et al., we partition the multivariate data set into *snapshots*, i.e., fixed time intervals with a stride of one. The encoder component reduces the snapshot with each succeeding layer of nodes until a bottleneck is reached, which forms the latent space of the representation. Symmetrically to the encoder, the decoder takes input from the latent space and, with succeeding layers, reconstitutes the original amount of dimensions. The reconstituted snapshot is then compared with the original snapshot using the mean square error (MSE). The quality of the training is expressed by the *reconstruction loss* (i.e., MSE) and the *entropy loss* (here, the Kullback-Leibler divergence). Minimizing those losses using a gradient descent allows the autoencoder to learn the multidimensional structure of the data set, encoding it in the



(a) projection quality decline over SD per parameter



(b) frequency histogram at SD 0.1



(c) frequency histogram at SD 0.9

Figure 4: Extract from the measurements of our dimension reduction evaluation. (a) The plot shows the influence of variation of model parameters with increasing amounts of standard deviation (SD) on the x-axis for MDS and PCA. Visually speaking, SD is the degree of deviation around an ensemble of randomly chosen centroids. The y-axis shows the percentage of maximum remaining variance not embodied by PC1 and PC2 for PCA, and the percentage of maximum stress induced by MDS. Varying frequency is the decisive factor for projection quality with both the methods. The frequency histograms at SD 0.1 (b) and SD 0.9 (c) over $t/2$ are shown for reference.

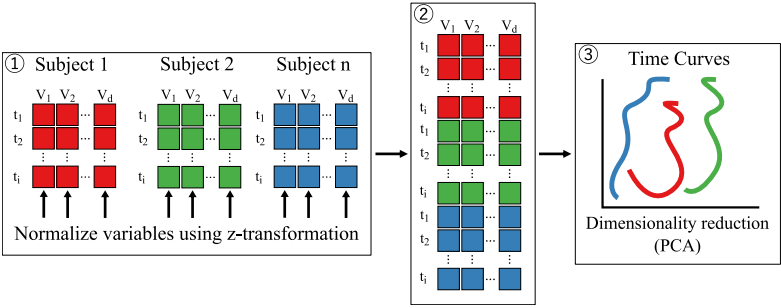


Figure 5: Schematic of the normalization and PCA projection for multiple subjects. ① Subjects are normalized individually using a z-transformation. ② The data frames are concatenated and ③ PCA is subsequently performed on this concatenated data frame ensuring identical loadings allowing comparison of individual subjects.

latent space. Due to the layer architecture, an autoencoder can represent non-linear structures, which contrasts it to PCA. Training means fitting the entire data set in this context. Accordingly, the resulting latent space represents the entire data set as a set of variables that contain a notion of compressed and simplified (dis-)similarity of the original data points. Since our bottleneck is variational, our latent space models Gaussian distributions, i.e., mean and standard deviation. This allows compensating for a certain amount of fuzziness, making the training process more robust. To obtain a visualization from this latent space, we employ PCA. Note that adding a CVAE does not improve results per se: convolution is a great tool for analysing oscillating signals but has to be taken with a grain of salt when applied to variables without meaningful spatial relationship [281]—even though there can be correlation. Thus, we apply the convolution only on the temporal axis. In summary, our application offers three different DR methods to allow for three different analysis paths: PCA maintaining global distances and relations, CVAE in combination with PCA to leverage its non-linearity to accommodate for highly complex regions in the time curve, and UMAP for cluster analysis. The ensemble analysis (**R5**) requires a collective DR of multiple subjects. Here, we apply z-transform individually to each subject prior to DR. The DR is then performed for all subjects at once, allowing for comparisons

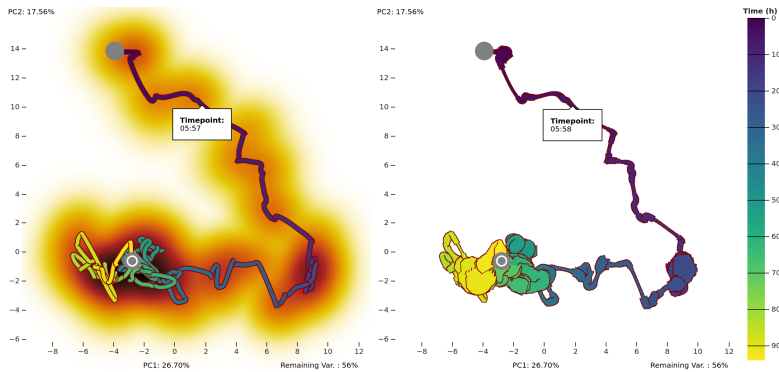


Figure 6: Time curves for subject VNA004 with constant width and color-coded density (left), and with data point density in proximity encoded as width (right). The curves are color-coded according to the time since the start of the measurement. Both curves are filtered using the average of a moving window. The background is colored by point density (high to low).

of subjects (Figure 5). Consequently, and for a meaningful comparison, the same set of dimensions must be used for each subject.

Time Curve visualization

Filtering The DR algorithms embed each multi-dimensional data point of the data set into a two-dimensional space. To obtain a time curve, these 2D data points are connected in chronological order. Since all DR algorithms emit rather cluttered time curves for real-world as well as generated data, it became clear that we had to reduce the complexity of the curves to cope with visual clutter. Therefore, we smooth the emitted data using a Bartlett-window moving average. The width of the moving window determines the degree of smoothing, which can be adjusted by the user. Obviously, filtering introduces a loss of information (R4). Hence, we devised different ways to compensate for this loss of information. One option is to encode the density of data for a given region before smoothing with color (Figure 6 left). We use a Gaussian kernel in conjunction with kernel density estimation to generate a smoother density visualization. This map allows assessing how many time points lie in the proximity of a time curve segment and, thus, enables the viewer

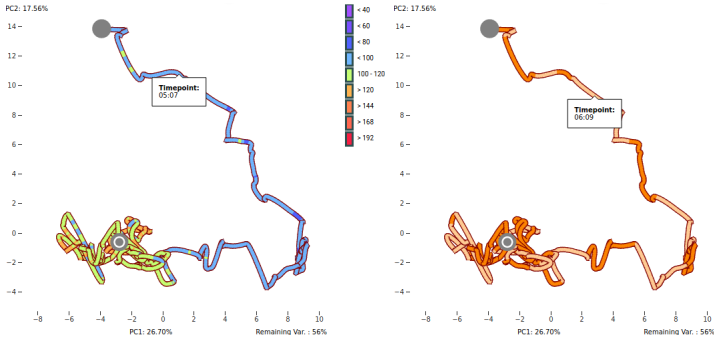


Figure 7: Left: Expected value coloring shown on systolic blood pressure. User-supplied expected value ranges for the variables are used for coloring. Right: Change point coloring shown on heart rate. color change (orange) illustrates that the preceding and succeeding intervals are substantially different for the chosen variable.

to draw conclusions about the velocity of changes in the system. The other option is a mapping of variables onto the width of time curve itself (Figure 6 right).

Space-Time Cube Time curves often self-intersect, which can make it difficult to follow the course of the curve. To remedy this problem, we also render the time curve in a space-time cube. Space-time cubes are a common representation in geospatial visualization to show the temporal progression of a path on a map by using the third dimension to display time. However, while the space-time cube helps to alleviate this specific issue, it also has inherent problems [14]. Thus, we only use the space-time cube as an auxiliary view alongside the planar time curve to support getting a more complete understanding of the state progression through the 3D visualization (see Figure 11). To improve spatial perception, we render the curve in the space-time cube as a shaded tube and add a grid as well as a shadow of the curve on the bottom plane. Temporally evenly spaced perpendiculars originating at the curve further add to the spatial perception.

Color Mapping We use the color channel to map additional data values in both planar and 3D time curve representation. Two ways of mapping

the elapsed time to the curve were implemented: one showing the time using a sequential color scale, and one using a discrete quantized scale. Two further color modes map derived values to the time curves that enhance the analysis (see Figure 7). In the first mode, the user can supply a value range per variable (**R2**), which represents the expected—or desired—value range for a healthy subject. The time curve can be colored per variable, with upward and downward deviations displayed in discrete percentage increments. Alternatively, a combined mode for all variables is available that colors each time point according to the variable that exhibits the strongest deviation. The second additional color mode is a change-point-based coloring. At change points, the trend of a variable becomes different from earlier time segments. This allows for a direct comparison of the overall patient state and whether an observed change in overall patient state coincides with a change point in a given variable (**R1**, **R2**). The change points of each variable are calculated using the *Ruptures* package [269].

Time Curve Boxplots and Scarf Boxplots

When plotting multiple time curves in a single visualization, the result is often dominated by visual clutter even for relatively low numbers of curves. Especially the overlapping curves are challenging to assess, rendering it difficult to see which one of the curves is representative. Therefore, we extract a representative hull that encloses a centrally-outward ordered share of curves. To establish this order, we resort to a class of order-statistical metrics called *data depth* [58] that describe how central an element is within an ensemble. In this context, centrality can be understood as representativeness. We use a metric called *functional band depth* [163], which builds on functional bands. A functional band B_k consist of $k \geq 2$ functions $f_{ir}(t \in I)$ with restriction interval I :

$$B_k \left(\begin{bmatrix} f_{i1} \\ \vdots \\ f_{ik} \end{bmatrix} \right) = \left\{ (t, y) : t \in I, \min_{r \in [1, k]} f_{ir}(t) \leq y \leq \max_{r \in [1, k]} f_{ir}(t) \right\} \quad 1.$$

A function lies between other functional bands if it is enclosed by the respective minimum and maximum values, as shown in Figure 8.

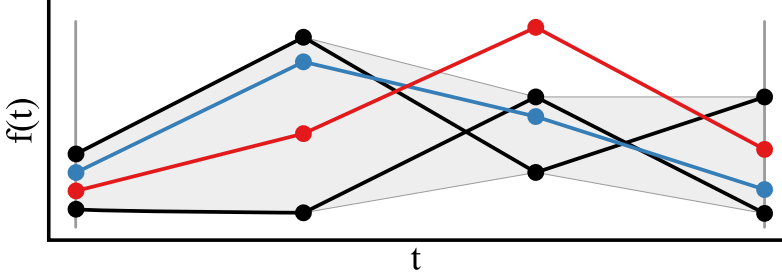


Figure 8: Schematic of functional band depth. The band (■), indicating the most representative functions (bounding functions, ■). The blue (■) function is included, i.e., belonging to the group of most representative function, whereas the red (■) one is not included, and thus is less representative for the ensemble of functions.

Formally, we can count this condition using an indicator function χ with $true \mapsto 1$ and $false \mapsto 0$ to define band depth $D_{n,k}$.

$$D_{n,k}(f_i) = \sum_{j=2}^k \binom{n}{j}^{-1} \sum_{l=1}^j \chi \left(\left\{ (t, f_{i(t)}) \mid t \in I \right\} \subset B_l \left(\begin{bmatrix} f_{i1} \\ \vdots \\ f_{il} \end{bmatrix} \right) \right) \quad 2.$$

This formula first counts how often a function f_i is contained in bands of length l . The number of actual inclusions is then put in relation to the possible ones. The process is repeated from bands of length 2 to up to bands of length k . A smaller k is more sensitive to variation, whereas a larger k is the more resilient to small fluctuation. Note that Equation 1 can also be expressed as a convex combination, which emphasizes the enclosing nature of the isosurfaces that can be defined on this metric, i.e., our enclosing hull. While data depth has been used to reduce visual clutter and design boxplots for quite some time [178, 264], it has not been applied to dimension-reduced curves yet. In this application, the calculation of the functional band depth is performed on the high-dimensional, z-transformed data. The enclosing hulls are then generated, according to the order established by data depth, from smoothed time curves after DR. Figure 9 shows our time curve boxplot that covers the 50% and 80% quantile of most representative time curves, allowing the omission of all time curves included in this quantile. Consequently, recognizing which

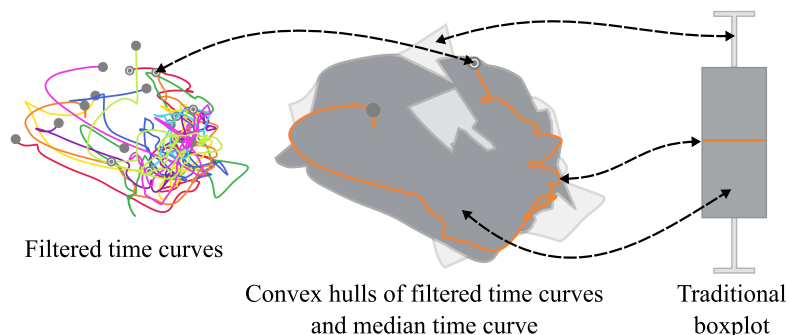


Figure 9: Visual metaphor of the boxplot as applied to time curves. The cluttered view of eight time curves is simplified by representing curves as areas corresponding to different quantiles: The 50% quantile (■) corresponds to the box of the boxplot, while the 80% quantile (□) corresponds to the whiskers. The orange (■) time curve is the most representative one (by functional data depth), and thus corresponds to the median line in the boxplot.

curves are outliers and which ones are representative for the whole ensemble is easily possible. Figure 9 shows this concept applied to an ensemble of eight curves. By showing the two quantiles (50% and 80%) and the most representative curve (which corresponds to the median), we follow the well-known design of a standard boxplot.

As the containment of a curve in the box area not only depends on being at the correct position, but on being at the correct position at a specific point in time, it becomes difficult to judge when a given curve is not included in the curve box area. Hence, we provide an accompanying scarf boxplot (Figure 10) that depicts box inclusion over time, with dark grey and light grey markers indicating the inclusion in the respective quantile. Since the scarfs are temporally aligned, it is easy to judge when a curve is representative and or not in comparison to other curves.

Details Using Small Multiples

The purpose of the time curve visualization is to give an overview of the progression of patient state (**R1**). While it allows for identifying potentially interesting points in time, it is not intended for a detailed analysis of individual parameters (**R2**). Given the unavoidable projection



Figure 10: Scarf boxplot illustrating representativeness over time from left to right for each subject that is not included in the inner quantile box plot area. colors and labels correspond to the ones in the main time curve boxplot, while the dark grey (■) and light grey (■) markers indicate inclusion in the respective boxplot area.

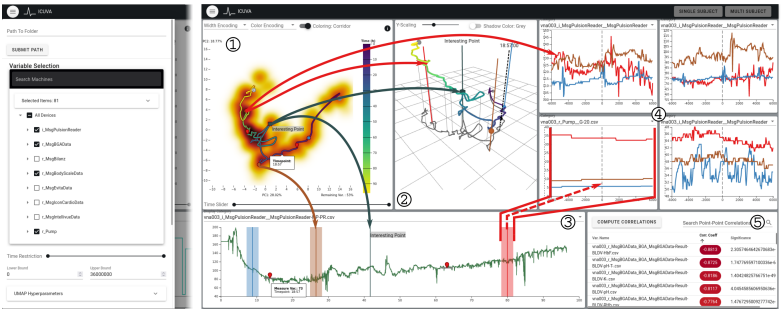


Figure 11: Overview of the single-subject mode (right) and expandable menu (left). The time curve shows patient state progression (top left; 2D time curve plot ① and space-time cube ②). An auxiliary line plot shows a single time series, with time in hours ③. Three samples are selected (■, ■, and ■) to showcase linking. Dark grey lines and markers with custom labels are used to annotate interesting point in time (e.g. peaks in specific measurement variables). The small line plots ④ shows superimposed time intervals (in seconds) based on selected points, for one selected variable each; the width of the bar in the auxiliary line plot corresponds to the, identically colored, time segment in the small line plots. The shown variables can be selected freely from the variables used for projection. The correlation table ⑤ compares two points in time that are indicated as red markers in the auxiliary view and the 2D time curve.

error, loss of information, and the sanity checking requirements (R4), we show details on demand in small multiples of unprotected data dimensions. Thus, an auxiliary line plot that depicts values over time and several small line plots allows for easy superimposition of small time-spans. For each line plot, the user can select a different data variable in a drop-down menu. Users can select between 1 to N points on the time curve or auxiliary plot. Each selected point corresponds to an identically

colored polyline in each of the small line plots. Respective time series are centered to the selected time point, showing backward and forward temporal development of the data (Figure 11). The user can select the time delta that is used for the abscissa. The temporal interval is also drawn as a semi-transparent window in the auxiliary line plot (Figure 11). This allows the user to get an impression of temporal change in individual dimensions by comparing the values at different points in time. It also allows for comparing the progression of different data dimensions and, consequently, to infer interdependency and correlation. The line plots relate to requirement **R2**, since they facilitate the individual inspection of each line plot while the small multiples view also allows for investigating localized trends and patterns.

4.2.5 Description of the visualization System

We implemented our visualizations described in Subsection 4.2.4 as a client-server web-based application. Costly calculations are performed server-side while the presentation of the results is facilitated client-side. This enables access from different terminal devices—a scenario commonly found in a hospital setting. The server was implemented using Python. The Flask framework [229] was used to communicate with the HTML5 client page. For data filtering and sampling, we use Pandas data frames [173]. Keras [47] and Tensorflow [2] were utilized for the CVAE, while scikit-learn [207] was used for PCA. The client-side visualization and UI was built with the Vue.js framework, and the D3 library [29]. For the WebGL-based rendering of the 3D space-time cube, Three.js was used. The system features two distinct views: the first view is meant for analysing a single subject and offers features for an in-depth post-hoc analysis. The second view offers comparative visualizations for an ensemble of multiple subjects Figure 11 shows the single-subject mode. The collapsible menu to the left is used to select a data set and configure most of the visualization pipeline including data selection, processing, and choice of DR technique. This mode features a patient state progression visualized as a filtered time curve and space-time cube visualization, a large auxiliary line plot, and four small selection-relative line plots. A drop-down menu above the time curve visualization allows

for the selection of a variable that is encoded onto the width of the time curve. Alternatively, constant width and spatial point density can be chosen. A second drop-down allows the user to choose between the color modes explained in Subsubsection Time Curve visualization. The filled circle (●) indicates the starting point of the curve, the dotted circle (⦿) the endpoint. To scrutinize the progression of individual variables, a large auxiliary line plot is shown below the filtered time curve. The visualized variable can be chosen in the drop-down menu above the plot. Right of the time curve visualizations, small selection-centric line plots are shown. All plots are linked views that support hovering and selection as well as annotating time points with labels. Selecting points on either the time curve or in the auxiliary line plot will determine the point in time that is shown in the small selection-relative line plots. These plots show the temporal surroundings of the selected time points over the chosen variables. In addition to the four small multiple line plots, a second way to compare segments of the curve was implemented: a pair of time points on the curve can be selected (red markers ♥ in Figure 11). The Pearson correlation coefficients are calculated for each variable of the two selected points over an equally-sized window (**R3**). The coefficients are listed in a table, which can be sorted by correlation or significance.

The multi-subject mode has an analogous menu for data loading and selection, with additional inputs for the quantiles of the time curve boxplot computation. The main view is a time curve plot showing an ensemble of (filtered) time curves as well as the time curve boxplot. It supports toggling the visibility of individual ensemble members and the enclosing curve boxplots (Subsubsection Time Curve Boxplots and Scarf Boxplots). To further inspect a specific subject, the user can switch to the single-subject view of this subject by selecting it in the legend. Since the superimposed time curves depict the similarity of the samples instead of a temporal alignment, we added a scarf plot below the time curves to illustrate box inclusion over time. It only shows subjects that are not included in the inner quantile range. Thus, representativeness can be analyzed both in terms of values and time.

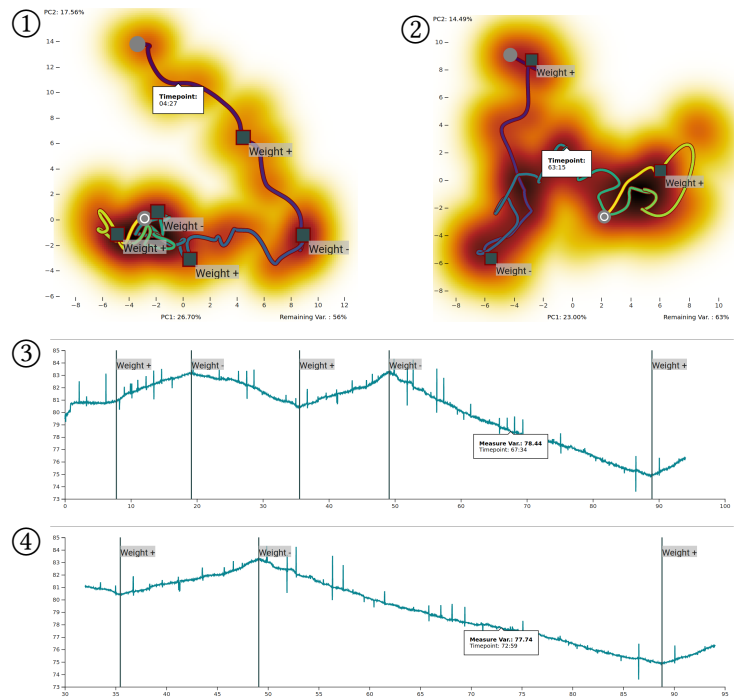


Figure 12: Juxtaposition of the time curve and the net-weight of the subject over the whole 94 h ① and shortened to the last 62 h ② and their respective auxiliary time-series line-charts (③ and ④). Time curves smoothed were with a window size of 150 (Figure 6 shows the same curve with a windows size of 50).

4.2.6 Results and Discussion

In this section, we detail several examples and investigations that show-case that our presented visual analysis approach is a promising tool for medical research concerned with ICU data. We also highlight how these use cases relate to the user requirements. For the single-subject mode, a set of 82 parameters from the data described in Subsection 4.2.2 was used. For the multi-subject mode, we present the results for nine animal subjects (one animal was omitted due to short experiment duration) using a set of nine relevant parameters each. We also evaluated the applicability of our approach to other ICU data sets using publicly

available data from PhysioNet [95]: MIMIC-III-Waveform [125, 180] and MGH/MF-Waveform [292]. We also investigated the potential of our visual analysis approach by gathering feedback from our domain experts during development and by conducting structured feedback interviews with independent medical professionals.

Single-Subject Mode

Figure 12 shows the filtered time curve of subject VNA004. The most prominent feature is the significant and rapid change in status during the first 20 hours, indicated by the comparatively small temporal change as shown by the small change in color and the large difference in state (patient stability: **R1**). While it is difficult to pinpoint, our medical experts have identified this segment of the curve as the onset of anaesthetics and the implantation of measuring devices that put a strain on the homeostatic system of the animal. Using the auxiliary plot and the small multiples (**R2**), we could identify several variables (e.g., heart rate, glucose, and blood pressure) that might explain this rapid change in the time curve. The dense region at the end of the time curve signifies that the patient state does not change as drastically anymore but only fluctuates slightly around a stable state.

During the VNA study, all subjects were infused with high amounts of liquids over an interval to be then deprived of liquids. This change can be seen in the continuous weight monitoring. The points at which the administration of liquids is either started or stopped leads to noticeable bends in the time curve. As shown in Figure 12, three of the five administration changes are in a hairball-like region, roughly between hour 32 and 94. In these 62 hours, the observed subject reaches a stable state, indicating that the variables do not change considerably. This stabilization can also be seen in several variables (e.g., heart rate, glucose, and blood pressure).

To disambiguate the hairball, the time curve was restricted to this temporal window of 62 hours (see Figure 12, right). The aforementioned bends in the time curve can be narrowed down to the fact that the amount of liquids excreted from or added to the organism are substan-

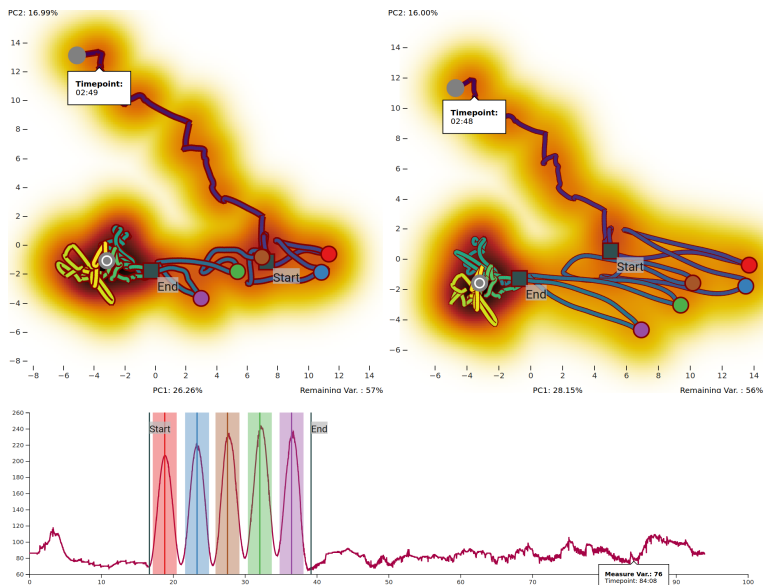


Figure 13: Artificial oscillations induced in 8 (top left) and 16 (top right) of the 82 variables of VNA004 lead to loop-like patterns in the resulting time curve, while the rest of the curve changes only marginally. This test shows that time curves are a robust and intuitive approach for the analysis of time-varying multivariate data.

tial (2.7 kg and 8.3 kg), straining the organism. Thus, it is expected that the state as a whole—represented by the time curve—will show a noticeable reaction (**R2**). To test our approach more systematically, we enriched the VNA data set with patterns corresponding to—exaggerated—scenarios encountered in real life. One such scenario is the sudden and repeated change of a few variables, e.g., a rapid increase in heart rate. A sudden change in some variables should lead to deviations of considerable magnitude at the corresponding positions. This is simulated by enriching an interval in a subset of values with sinusoidal functions (Figure 13, bottom). These should produce periodically similar patient states resulting in periodically similar positions in the time curve (i.e., loop-like structures). The peaks induced by the oscillations should indicate a substantially altered overall state, leading to noticeable deviations from the original time curve trajectory. This effect can be seen

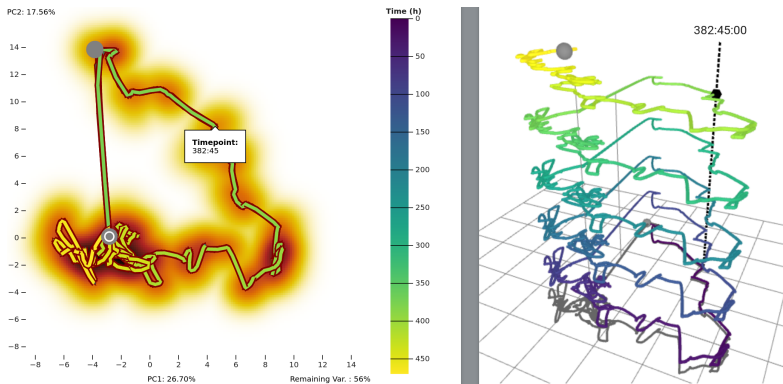


Figure 14: VNA004 data set repeated five times. While the 2D time curve exhibits heavy self-occlusion leading to a loss of information, the space-time cube clearly shows the temporal progression of the five identical parts of the concatenated data.

in the marked segments in Figure 13, which also shows that the effect is amplified when increasing the number of variables that are simultaneously enhanced with sinusoidal patterns. Another scenario concerns repetitions in patient state, e.g., large-scale cyclic behavior. To simulate such behavior, a data set was replicated and concatenated, so that it contained multiple identical copies of itself. DR was performed as usual. As expected, identical patient states are plotted at identical position, and thus the expected time curve exhibits cyclic patterns (Figure 14).

To demonstrate the applicability of our approach to other ICU data with different, more general characteristics, we applied the single patient analysis to data from PhysioNet [95]: the MIMIC-III waveform data set [180] and the MGH/MF waveform data set [292], which are openly available measurements of typical parameters recorded in daily clinical routine. While the MIMIC-III data set spans longer time-frames (long-term intensive care), the MGH data set spans only shorter time-frames. This displays the ability of our presented application to handle different types of data, especially with regard to varying length of the time series.

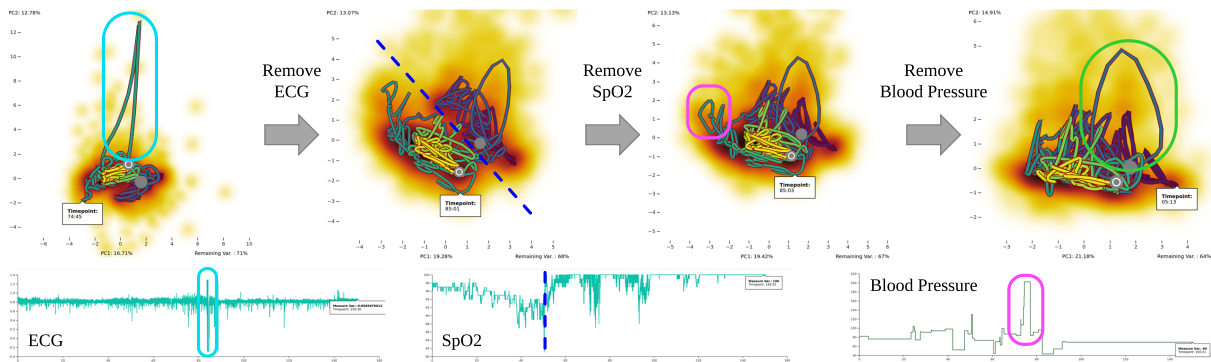


Figure 15: Analysis of a patient record from the MIMIC-III data set using our application (patient: p042075, 15 variables). The time curve exhibits a strong peak (■) caused by a peak in two out of three ECG recordings (one is shown below). After these two variables are excluded, the bifurcated structure is revealed more clearly (■), which is at least partially caused by the oxygen saturation, as the bifurcation becomes less prominent after removing the SpO2 variable. Lastly, two loop structures are visible (■, ■). While one (■) can be removed by excluding two variables containing blood pressure data, the last one (■) seems to be induced by multiple variables, warranting further inspection.

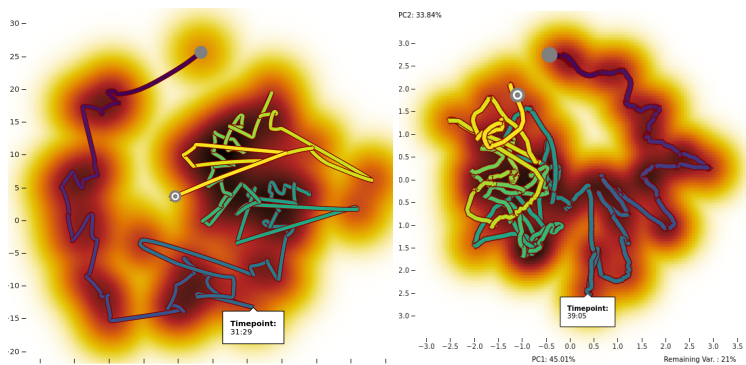


Figure 16: Time curves for VNA004 created using UMAP (left) and our CVAE model coupled to PCA (right). See Figure 6 for PCA.

Figure 15 showcases an exemplary analysis of a ~ 150 h patient record from the MIMIC-III waveform data set using our presented application. When all 15 variables are included in the projection, a prominent peak can be seen (Figure 15, ■). Using the auxiliary plots, we explored the variables and found possible reasons in two ECG records. After these records were removed, this peak is no longer present, supporting the hypothesis that these two were the reason for the peak. Several other features now become more prominent: a bi-partition of the whole structure, and a deflection of the time curve (■ and ■ respectively). We analyzed these features in the same way as the first peak, revealing oxygen saturation (SpO₂) and blood pressure as the main candidates for these features. Removing the oxygen saturation led to a less pronounced partition of the curve, while removing the blood pressure removes the loop. Afterwards, a last prominent loop-like feature can be seen (■), which, however, can not easily be related to a single variable but instead traced back to a combination of multiple variables. This could indicate a medically significant, complex process, warranting further exploration.

We also compared time curves generated using only PCA (Figure 6) to ones using CVAE coupled to PCA and UMAP (Figure 16). The most notable feature is the change in the representation of larger distances. While PCA correctly identifies the first few hours as a phase of drastic

change, it does not show the smaller changes in the later phase particularly well. CVAE as well as UMAP also correctly identify this early trend, however, due to their non-linearity, scale it down. This allows for a higher resolution of the more fine-grained changes in the later stage. While this leads to a more pronounced visualization of smaller changes when using the CVAE, the linearity of PCA might make it easier to judge global similarities. UMAP typically generates more pronounced local clusters along the time curve compared to CVAE. Hence, the users can choose between the methods in our application, based on their preference.

Multi-Subject Mode

As shown in Figure 9, the time curve box plot allows the user to assess the representativeness of the subjects (**R5**). Figure 17 shows the multi-subject view, where each time curve represents one of the nine animal subjects. Six of these nine subjects (Figure 17b) were part of a temperature control experiment [212]: ■ and ■ had no active temperature control management, ■ and ■ had their temperature adjusted manually by ICU staff using a hot air blanket, ■ and ■ were subject to automated temperature control using a hot air blanket. As subject ■ had a reduced experiment duration of 57 hours [212]—compared to 69 hours for the remaining subjects—the analysis is also restricted to this 57-hour window. The group for which the temperature was not actively adjusted (■ and ■) exhibits rather different progressions of their respective time curves (Figure 17c). The time curves of the manually adjusted group (■ and ■) show a more similar trend (Figure 17d). The progression of the subjects with automatically adjusted temperature (■ and ■) is the most similar (Figure 17e). Although temperature is only one of the nine included variables, it seems that the subjects can be grouped by the applied temperature adjustment. Actively adjusting the temperature may improve homeostasis management, which is in turn visible in the time curve representing the organism's state.

The feasibility of the multi-subject mode was also tested on a subset of the MGH/MF waveform data set [292] consisting of 16 patients (*mg*h002–

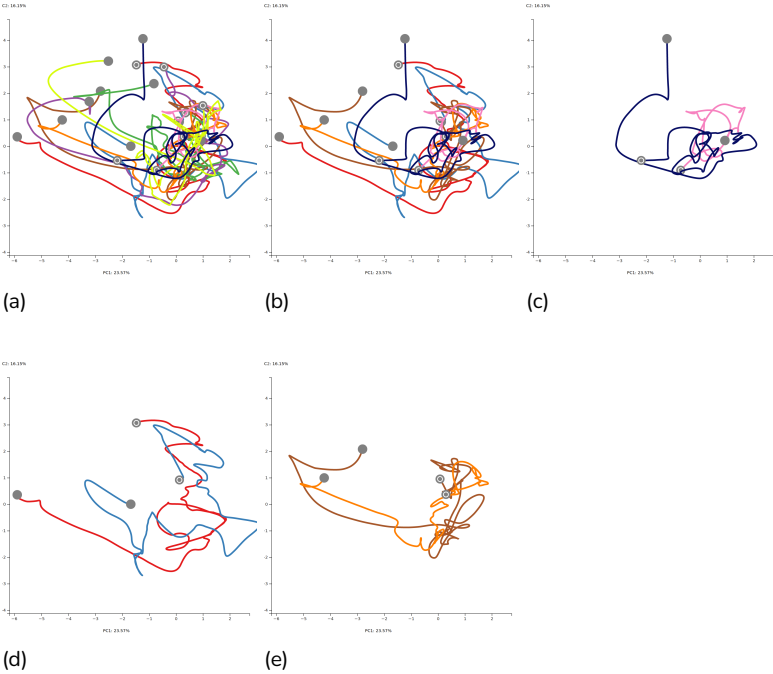


Figure 17: (a) An ensemble of nine animal subjects shown as filtered time curves. (b) The six subjects that were part of the temperature control experiment. (c) Sub-jects ■ and ■ : no active temperature control. (d) ■ and ■ : manual temperature control. (e) ■ and ■ : automated temperature control. Each time curve results from the same nine parameters of each subject.

mgh021, with *mgh012*, *mgh014*, *mgh017*, and *mgh018* being omitted due to missing variables). As the data contained events assumed to be application and removal of the measurement devices, heavily skewing the projection, the tool’s functionality to restrict the time series was used to cut off the beginning and end of the time series. When displaying the complete time curve ensemble, cluttering is prohibiting any analysis (Figure 18, left). Our proposed time curve boxplot visualization can be used to declutter the visualization significantly, while still retaining some information, like the identity and shape of the most representative curve (Figure 18, center). Inspection of the outliers—time curves that are not

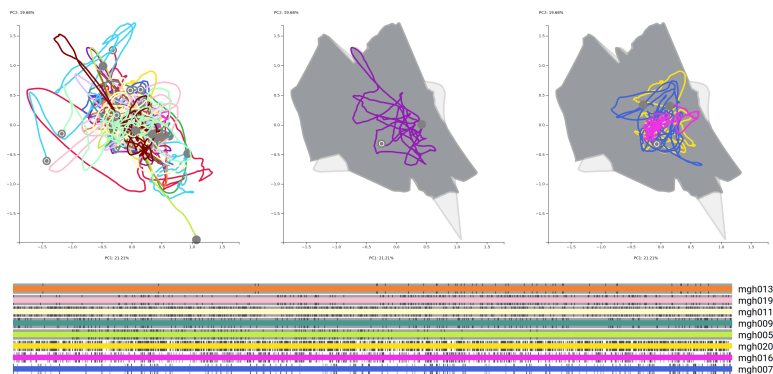


Figure 18: Ensemble of 16 patients from the MGH/MF data set. A naïve overview of all time curves is quite cluttered (left). Our time curve boxplot (Center: combined with the most central time curve; right: only outliers) allows for detailed analysis based on functional data depth. Bottom: Scarf boxplot showing detailed information about the inclusion of the curves for the respective quantile hulls.

wholly contained either of the percentile ranges—shows that not only spatial but also temporal inclusion are of relevance when calculating the functional data depth (Figure 18, right). The scarf boxplot below visualizes the true extents of inclusion in the respective hulls, both temporal and spatial. It reveals that the yellow curve showing subject *mgh020* is quite often within the 50th percentile range, but only for very short durations, which might be an interesting aspect suggesting a more detailed inspection of the individual variables. The orange curve representing subject *mgh013* is always in the 80th percentile range but only rarely in the 50th one, suggesting that this patient might be overall more dissimilar to the most representative patient, but has no strong occasional deviations like *mgh020*.

Expert Feedback

To estimate the utility in clinical practice, we collected feedback from our expert co-authors as well as four medical doctors from the local university hospital. The latter were neither involved in the development of our application, nor had any other contribution to this work. They saw our tool for the first time in the feedback session. The feedback

Table 1: Questions asked at the end of the structured feedback sessions.

How is such data currently analyzed? Which programs or which workflow exists so far? Would the tool help with such data?
Does the tool help you to make a statement about the stability of the patient/subject? If yes, how (which views/visualizations)?
Are there functionalities or visualizations that helped you in solving the tasks? What aspects did you enjoy?
Are there functionalities or visualizations that were not useful to complete the tasks? Were there aspects hindering you in completing the tasks?
Do you think that the tool would help you to get more details about the different readings/vital parameters at one or more time points starting from the overall condition of the subject, or rather to learn more about the overall patient condition starting from one specific aspect/measurement at one time point? (Or both?)
As initially mentioned, noise, measuring errors, and outliers are filtered in the beginning. Do you agree with this approach?
Is it good to see exactly the measured data in the plots, that is, to see the last available measured values at each point in time? Or would an interpolation of the data be better so that one sees a smooth curve?
What additional features or plots would be helpful? Is there anything else you can think of?

from these experts was collected in structured feedback sessions of about one hour each, in which they were given introductions, performed three tasks, and were asked eight questions. During the tasks, we used a think-aloud protocol. In the first task, the users were presented with a projected time curve with two distinctive points selected and were asked to find possible reasons for this behavior, essentially performing a top-down workflow (R2). The second task was set up for a bottom-up workflow, with a distinctive point being selected in a single variable using the auxiliary line plot. Finally, the users were presented with the multi-subject setup described above and were asked to evaluate if the temperature groups defined by the experimental setup can be validated by the time curves course.

The eight questions were designed with the requirements in mind to get a measure of the general usefulness of the tool (see Figure 19 for screenshots of the prepared starting points for each task and Table 1 for the questions asked).

Overall, the users clearly understood the presented time curve as a representation of the overall patient state and its progression. All users mentioned its primary benefit in the use as a patient overview, as the

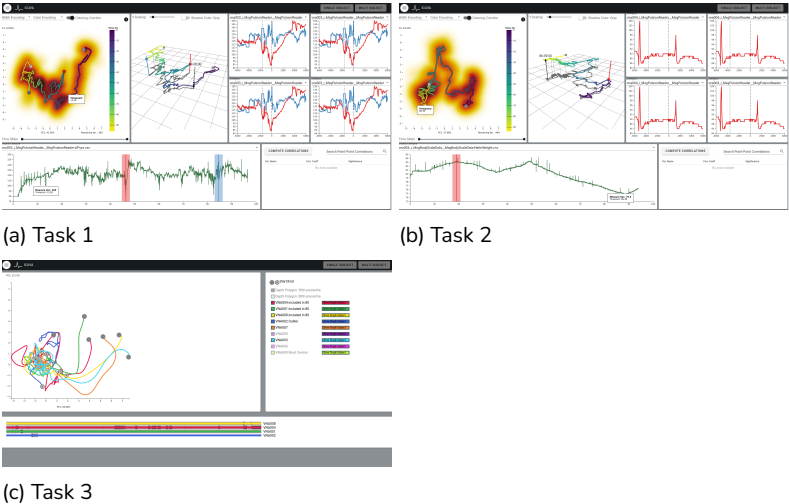


Figure 19: Initializations of our application for the three tasks executed by the medical domain experts during the structured feedback session.

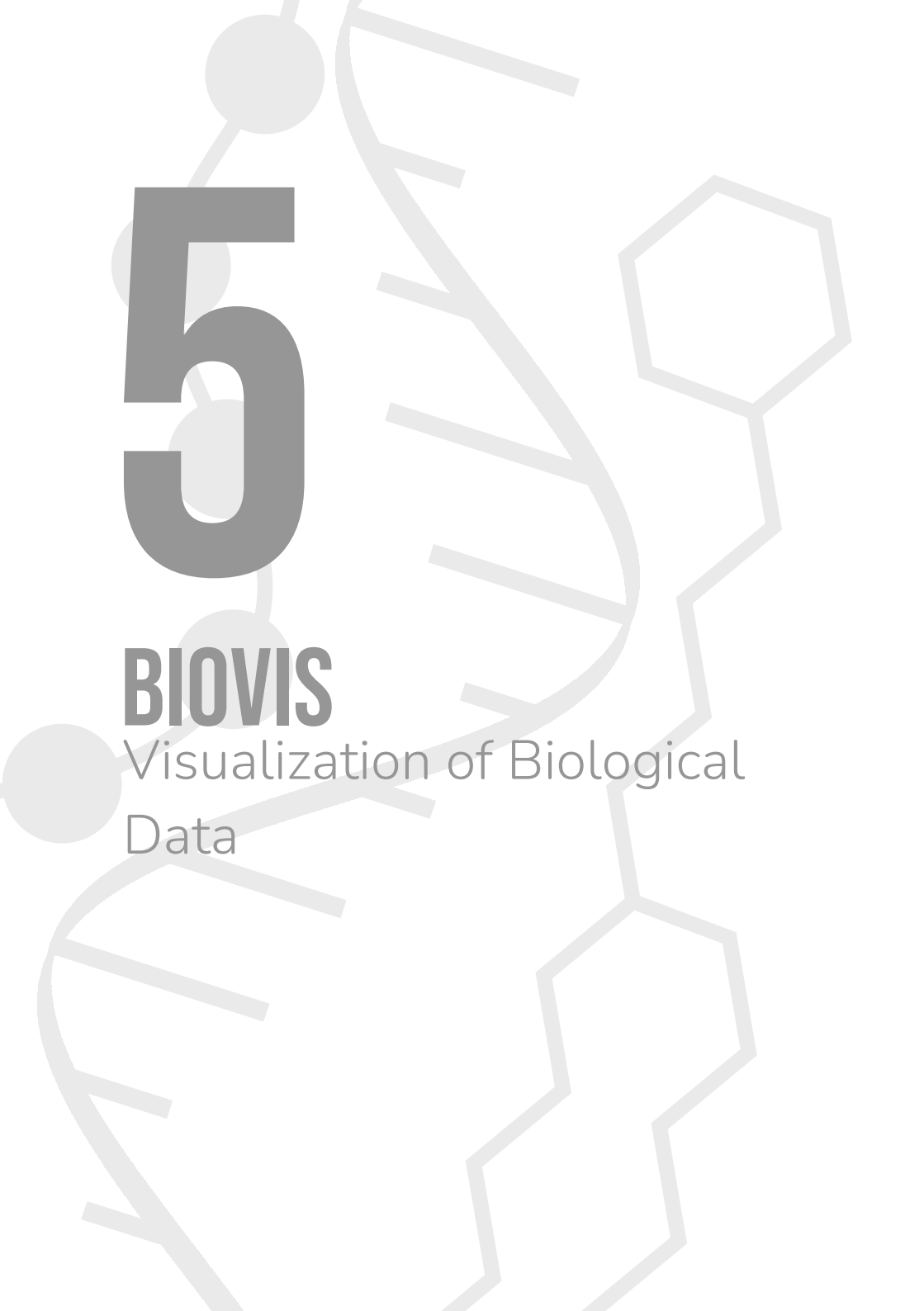
tools and workflows mentioned as current clinical practice (e.g. Excel, SPSS) mainly utilize single parameters, often reduced to a few statistical key figures (i.e., mean and median), making it difficult to judge the overall patient state. One of the independent experts explicitly noted that such a visual analysis tool is “*exactly what we need*”. They rated the combination of detail plots (auxiliary and small multiples) and time curve as highly usable to judge the overall stability of a patient (**R1**). While the users understood the general concepts of the DR methods, several of them pointed out that—in part due to the complexity and unfamiliarity of dr—some training is needed to fully exploit the potential of the system, especially if it was to be used in a live environment. All users were also successful in judging the stability of patients when used in a multi-subject scenario (**R5**). One expert noted that, due to time constraints and the large amount of generated data, retrospective analysis rarely happens in clinical practice, although it would be important. They reckoned that the fast overview provided by our application—especially the time curves—could enable more frequent retrospective case analyses, which they

expect to be highly useful. The top-down approach was favored by most users. However, all users were also successful in using the bottom-up approach. One user even preferred this way of working, indicating the functionality of both approaches (**R2**). When asked about the handling of missing data, the users generally either had a strong preference for the chosen method (Subsubsection Data Cleansing and Preprocessing, **R4**), or did not have a clear preference, mentioning pros and cons for either: our approach and smooth interpolation. Furthermore, clearly communicated and user-controllable filtering was strongly preferred as well, as even values perceived as “faulty” could yield helpful medical insights. One user noted that curve visualizations, especially as space-time cubes, are not commonly used for ICU data in the medical domain. However, they were shown to be beneficial for solving complex tasks in other domains [146]. In general, the additional auxiliary functions (i.e., space-time cube, correlation view, and color/width modes) were less frequently used. However, some of the experts explicitly noted that they consider them interesting and useful and presume that they would apply them more often after more extended use. Nevertheless, only two of them rated the space-time cube as not useful. Regarding possible additions, the inclusion of the value corridor in the auxiliary view (as already implemented by color-coding for the time curve) was mentioned multiple times. Multiple users reckon that it would be beneficial to allow for the display of multiple variables in the auxiliary plot. Some users expressed the need for a guide to more easily find variables that exhibit significant features for time points selected in the time curve. In conclusion, our application shows high potential and acceptance by medical experts, not only for the post-hoc analysis of experimental data, but also for fast and convenient retrospective analysis. Based on the expert feedback we collected, it seems to satisfy the increasing demand for analysis tools to cope with the increasing data output and complexity of medical surveillance devices.

4.2.7 Conclusion and Outlook

We presented a web-based visual analytics application for multivariate, time-varying ICU surveillance data. Due to the large amount of data, a

comprehensive view on the patient state progression is quite difficult and traditional visualizations like line charts fail to adequately present the data. Our application uses DR to visualize these high-dimensional data as time curves to provide an overview of this progression of the patient state over time. The time curves are shown as a 2D plot and in a space-time cube. Since ICU data can have missing or erroneous values and parameters with varying sampling rates, we extended the time curves using filtering and smoothing. To prevent misinterpretation of the smoothed time curve and to show the density of the input data, the underlying data can be shown as color-coded density in the background. A small multiples view shows details on demand about the individual dimensions as line plots and allows users to select and compare different points in time. Our visual analysis application also provides methods to facilitate the comparison of multiple patients by projecting their individual filtered time curves into one two-dimensional space. Furthermore, the system can depict an ensemble as time curve boxplots that summarize typical time curves while highlighting outliers. The time curve boxplots are complemented by a scarf boxplot that shows at which points in time the outlier curves deviate from the box. We evaluated our application with real-world data and structured feedback session with medical experts, who consistently rated our application as useful and see a definite potential for further developments. In the future, we also want to extend the scope of our application to live data, which would allow for monitoring of patients in the hospital. To this end, our application has to be improved and retargeted in several ways: first, it has to be ensured that the time curve does not change too drastically when adding new samples and recalculating the DR. One possible avenue is the adaption of projection techniques to allow for the progressive addition of data points. Several interesting approaches to avoid frequent, global recalculations have been proposed for streaming data (e.g. [89]). Improving projection quality, e.g., through landmark-based guidance as proposed by Vernier et al. [280], is an active field of research. Its transferability to our application requires further investigation.



5

BIOVIS

Visualization of Biological
Data

The biomedical domain spans an immense variety of topics. Research in those topics is fueled by an equally large variety of different experimental approaches. Consequentially, different techniques require different data sources, be it recording devices, like pulse-oxymeter, as it is the case in the previous chapter, or biochemical assays, resulting in different types of data.

The type of data source and the process of how the data is generated also influences how readily such data can be understood. As mentioned, EHR data sources often provide easily understandable values out of the gate, without any additional data-wrangling required. In contrast, biological data as found, for example, in *proteomics* is a lot more complex. To measure the proteome of a sample, a large amount of background information has to be understood. This includes what proteins do and how they are built, how they are measured and, in the end, how a meaningful quantity can be derived to be interpreted.

5.1 Fundamentals of Biological Data

To understand the more complex biological data types, the fundamentals of molecular biology need to be explained. In general, when we see a phenotypical effect, i.e. things that are noticeable to the *outside world*, for example, a hereditary disease, or a malignant tumor, it is the result of deviations in the activity of genes, known as *gene expression*. To unravel the term *gene expression* we first recall the definition of a *gene*. Genes are regions on the genome—as *deoxyribonucleic acid* (DNA)—that encode the information needed to perform cellular functions. Most of the time they can be (over-)simplified as *one gene one function* [231]. During gene expression, this information stored in an organism is translated and with intermediate steps generates bioactive molecules that affect the organisms on cellular or tissue level. Gene expression is, thus, the basis of all physiological functions, from supplying energy to our cells to regenerating skin cells when the skin is damaged. This means gene expression is also the molecular way of answering to changes in the outside environment. On the other hand, erroneous expression of genes can be detrimental to the organism as well. For example when mistakenly

5 promoting the growth of blood vessels, which enables cancerous growth of tumors [126]. Or when errors in the information of globin genes occur, which will cause hemoglobins, the oxygen carriers in our blood, to be malformed [102]. Besides pathological conditions the variances in expression, both what and how much is expressed also are responsible for natural variances in individuals. Eye and hair color, for example, are also dependent on how much melanin, a brownish-dark pigment, is generated and stored in the iris or the hairs, as an end product of gene expression. The different stages of gene expression, often subsumed as the *central dogma of molecular biology* [55, 232], are: Genetic information is stored in DNA itself, as genes. DNA is stored, densely packed, in the nucleus. The DNA acts as a template for other information-carrying molecules that can be distributed to other structures inside the cell. This process is the transcription of genetic information into *ribonucleic acid* (RNA). Here, the extent of how much and as a consequence of what a gene is transcribed is the key factor on how expression is influenced. RNA then is, most often, translated into amino-acid chains, which form proteins the fundamental actors of the cellular machinery. Many of these proteins are enzymes, and these enzymes change in which quantity, or if at all, specific molecules, like glucose, are metabolized for further use. To describe how expression, and importantly, which stage of expression is causal for a phenotype, multiple molecular biological disciplines have formed.

While this description of the process is integral, experimental biology needs techniques to measure those processes and molecules. These techniques are used in *Omics*, which describe disciplines, that measure and describe the makeup, the concentration, and often the changes in concentration of the major bioactive molecules [233]. Most often we consider four different *omics* types corresponding to the major stages of gene expression: *genomics*—which gave rise to the name *omics* in the first place, *transcriptomics*, *proteomics*, and *metabolomics*. In general, researchers try to determine composition and changes in or between one or multiple organisms. For *genomics*, this includes finding mutations like *Single Nucleotide Polymorphisms* (SNPs) [168], copy number varia-

tions [94] and general structural changes, or even the communal *pan-genome* of groups of organisms, be it from the same or from different species [282]. *Transcriptomics* measures the level and identity of transcripts in a sample [234]. Oftentimes this is done to compare different experimental conditions to ascertain in which situations a specific gene is expressed more or less intensely. In *proteomics* both, the level and the identity—by composition of amino acids—of proteins in a sample are determined [235]. As the protein level is closer to the acting layer on a metabolism level, changes in the *proteome* more realistically describe the molecular biological reality. Finally, *metabolomics* aims at directly measuring the type and concentration of metabolites in a sample [203]. This is especially interesting when the expression cascade is not well understood or very complex, as metabolites are the direct actors of expression in many cases.

This describes the basics of omics. To properly understand the challenges associated with them, however, both on an experimental level and on an analysis level, especially the uncertainties they induce, the next section will detail the history as well as the current state of the art in the specific omics. These sections are limited to the types of omics whose data was used in this work, that is *transcriptomics*, *proteomics*, and *metabolomics*.

5.1.1 Transcriptomics

As mentioned, with the term transcriptomics we describe the measurement of the entirety of molecules transcribed from the DNA. Primarily researchers are interested in coding RNA [263]. That is, RNA molecules that will be translated into proteins, which is called *messenger RNA* (mRNA). Besides, mRNA, there are several types of *non-coding* RNA, including, among others, *ribosomal RNA* (rRNA), *micro RNA* (miRNA) and *transfer RNA* (tRNA). These *non-coding* RNA perform various roles, rRNAs constitute major building blocks of the ribosome while tRNAs transport amino acids, both processes integral for the translation from transcript to protein. miRNAs on the other hand acts as an additional regulation of expression, this time acting post-transcriptional [204]. Due to this different roles transcriptomic analyses are performed for an equal

amount of varying reasons. For example, rRNAs is analyzed during phylogenetic research as well as to study the composition of prokaryotic communities due to its evolutionary conservation [30, 247]. Nevertheless, as mentioned, mRNA analyses are by far the most commonly performed [148].

mRNA is transcribed from the DNA and thus, a main intermediary of gene expression. After transcription and transport from the nucleus, mRNA is translated into amino-acid chains, most often proteins [232]. Thus, analysis of mRNA can be performed to assess the expression levels of genes, both in general as well as for specific genes or groups of genes. On the first level expression analysis is performed to prove the existence of expression of specific genes in specific conditions. However, often instead of only measuring the level of transcripts, differential expression analysis is performed. This means that the quantitative expression of the transcriptome—or of its subset—is compared across one or more conditions. Conditions are often experimental stimuli. In pharmaceuticals, it can be the treatment with a drug [166], in basal research it can be the application of an environmental stressor, like drought stress on a plant [154]. Expression values from conditions are then compared with control groups to estimate what influence the stimuli have on expression.

While reasons for the analysis of expression were probably present minutes after the discovery of what the purpose of mRNA is, the technical means needed to be and are still developed to accommodate the experimental designs. One of the first methods to detect RNA, and more importantly specific RNA, is the *northern blot* [7] a hybridization approach named after its earlier and more prominent cousin the *southern blot* [257], which is used to detect DNA. Quantification of the respective biomolecules using blotting is, while possible, difficult and labor-intensive, and not as sensitive as needed for some levels of mRNA [80] or DNA [240].

This drawback was reduced substantially with the invention of the *polymerase chain reaction* (PCR) [183] which—as known now even to most laypersons, due to the COVID-19 pandemic—allows the amplification, that is the copying of DNA molecules. PCR can only amplify specific

DNA sequences, if the sequence, or neighboring regions, are known. From this known sequence so-called *primers* are generated that act as a starting point for amplification on each strand. Originally, PCR could only amplify DNA [38]. However, enzymes that *reversely* transcribe RNA back into DNA—defying the *central dogma of molecular biology* [55, 232]—reverse transcriptases have been described in the 1970’s [16]. These enzymes allowed for the generation of *copy DNA* (cDNA) from RNA, and thus first quantifications of mRNA with PCR [20]. In brief, the actual quantification is then done using special fluorescently-labeled nucleotides, during the amplification steps, with the fluorescence signal corresponding to the amount of cDNA and in turn RNA. Additions to PCR that allow the precise quantification [115, 159] as well as the usage of thermostable enzymes for reverse transcription [246] gave rise to modern *realtime quantitative PCR* (qPCR). While qPCR can be very precise it is difficult to use it to measure a complete transcriptome with it, as a complete set of primers would be needed incurring high costs and laboratory work.

To tackle this, two other high-throughput approaches were developed. The earlier one is called *microarrays* [62]. Its basic idea is very similar to that of RT-qPCR. Instead of using PCR and measurements during amplification, however, a signal is read out from fluorescently-labeled nucleic-acids bound to specifically designed probes bound to a plate in an array shape [41]. These probes are specific for genes and herein also lies the limitation of microarrays as an assay. Back then sequencing and prediction of genes was time-consuming and expensive the generation of probes and arrays with custom probes was limited. On the flip side, if routine quantification is performed, for example in a medical setting, ready-made arrays help to bring down the cost. Still, complete transcriptome analysis “for the masses” was out of reach with microarrays.

While sequencing was primarily used to sequence genomes as a part of genomics, not for transcriptomics, all presented methods need to know the structure and of course the identity of the genes that need to be researched as a part of genomics. While there have been a variety of techniques to sequence, at this point the primary and most efficient approach

to sequencing is called Sanger-Sequencing [236]. Sanger-Sequencing is also called chain-termination-sequencing as it utilizes chain terminating nucleotides while amplifying DNA in a laboratory to generate fragments of different lengths. By specifically using one of the four specific chain-terminating nucleotides researchers then know the last nucleotide of a given fragment generated using these specific nucleotides. Sorting these fragments by size, using gel-electrophoresis, then allows to decode the original sequence.

This is a low speed process and has a high cost. With Sanger sequencing being the only viable solution, heavily limited qPCR, and microarrays. This changed with the advent of *Next Generation Sequencing* (NGS)-methods in the mid-2000s, as it pushed the *per-base* cost of sequencing to a level where it was feasible to quantify nucleic acids through sequencing alone in a high throughput manner. *RNA sequencing* (RNA-seq) [187, 287] was born. It has to be noted, however, that especially in the initial steps of the establishment of NGS-methods the costs were still, compared to today, high. Especially in this early phase, many companies vied to be the leader in this new NGS and RNA-seq-space [287], offering their take on NGS—a battle soundly won by *Illumina* [186].

Irrespective of the employed techniques, the general basic functioning principle of all early RNA-seq approaches is similar [287]. Like in the earlier approaches the RNA is transcribed to cDNA. Optionally, depending on what part of the transcriptome should be analyzed, different steps, like rRNA depletion, are performed beforehand. cDNA is then split into small fragments and ligated to adaptors. The totality of the modified cDNA fragments then constitutes the *cDNA-library*. The minutia of *library preparation* differs between the sequencing techniques. All techniques then sequence the library to create the expression readout in the form of fragments called reads.

One of the main drawbacks, of all of these early NGS-systems exhibited, is that only very small fragments (30-400bp) can be sequenced, and thus quantified [287]. The resulting reads can be mapped to a reference or, as they overlap, assembled to a transcript assembly, when performing *de-novo* transcriptomics. Consequently, the number of reads, mapped

or assembled to a transcript, can then be counted to quantify a given transcript's expression. The short lengths of these reads then mean, for both *de-novo* and reference-based transcriptomics, that mapping becomes more involved and potentially introduces uncertainty. Due to the short length a very high number of reads has to be generated to result in meaningful coverage, that is how often is a given base sequenced—which also gave these approaches the moniker *shotgun sequencing*. Mapping in itself provides numerous challenges like alternative-splicing, as one gene does in fact not have only one function.

Especially early on, error rates of sequencing, which is the wrong base being called at a specific position, also required the development of specialized quality control methods. In general, a wealth of different tools, like the *tuxedo* suite [268], was developed to facilitate the different steps of RNA-seq that can be combined to tailor-made pipelines fitting specifically to the intended analysis. Furthermore, reads can map to multiple positions or biases can over-/under-representation in certain regions resulting in unequal coverage [255]. Additionally, sequencing depth, meaning how many reads are generated, in proportion to the whole transcriptome, also dictates how sensitive, i.e. how easily lowly expressed genes are detected. Often researchers do not only observe absolute expression values but should compare expression between conditions. This is called *differential expression*, which is one of the most commonly performed transcriptomic analyses. For this, additional statistical techniques need to be applied, like normalization of read counts, before the comparison of the samples.

All the steps required to prepare the RNA for sequencing can also introduce uncertainty. The reverse transcription to cDNA can introduce biases [283]. Assembling the short reads, as already mentioned, can introduce additional biases. There have been considerable advancements in techniques that solve some of these inherent issues of *shotgun sequencing*. One such advancement is the so-called *Single-molecule real-time* (SMRT)-sequencing [71]. As the name implies, in SMRT-sequencing a single molecule of nucleic acid is sequenced without fragmentation. As there are protocols for both DNA and RNA sequencing this solves

multiple issues. For one the transcription to cDNA can be omitted and the long reads generated trivialize the mapping and assembly. However, SMRT-sequencing also introduced its own set of shortcomings. Initial approaches in the early 2010s suffered immense error rates (13% [220]) and low throughput. Only recently (2022) have the error rates caught up to *shotgun sequencing* standards [155]. Still, because of the recency of these developments, the absolute majority of transcriptomics experiments today are analyzed using *shotgun sequencing*.

5.1.2 Proteomics

A fundamental problem of transcriptomics is, that no direct estimation of proteins can be done based on RNA-seq, as not all measured RNA is translated into a protein. Proteins are, however, the main actors of cellular functions, and thus a target for many research topics. They are responsible for generating energy, with ATP-synthases, protecting us against infectious agents, as antibodies, cleaving molecules to be more readily processed, with enzymes, with a host of other basic functions like contracting muscles when moving.

These functions are governed by the structure of the protein. In general, we consider four tiers of structure. The first tier is the *primary structure* a sequence of amino acids, a family of small-molecules of different sizes and with different chemical properties [48]. On the second tier, proteins organize in *secondary structures*, structures like *alpha helices* and *beta sheets* that are three-dimensional and form due to the interaction of the amino acid residues [49]. Multiple of those chained *secondary structures* form *tertiary structures* [50]. In many cases, the *tertiary structure* is already the complete Protein, and thus much of the shape and function of a protein is defined by its *tertiary structure*. Some proteins combine multiple, disconnected, *tertiary structures* to form a *quartenary structure* [51].

The reasons for quantifying proteins are much the same as they are for transcripts, just at a different position in gene expression. In many cases, one is then posed to question why one should use *transcriptomics* in favor of *proteomics*, after all, *proteomics* is closer to the *acting layer* of gene

expression. This comes down to how the proteome is measured, both historically and currently.

Proteins, as a whole, have been quantified since the 1950s [160] using coloring agents that bind to a substrate followed by the measurement of coloring. However, these early types of quantification were unspecific and not sensitive even though later versions were more sensitive the main issue with inspecificity remained [31]. Another class of assays are immunoassays, in which antibodies bind to the proteins and then measured either using radioactive [97] or light-based methods [226]. A big drawback of those assays is, that although they are highly sensitive, specificity comes at a cost. Using non-specific antibodies does not allow for the easy quantification of specific proteins, while the development of specific antibodies is very cost and time-intensive. This drawback can be somewhat mitigated by employing another technique: gel-electrophoresis [140].

At its core gel-electrophoresis is a chromatography-technique, i.e., a technique to separate mixtures, such as a protein mixture [140]. Several properties of the proteins can be used to separate them, most commonly size or the isoelectric point. Using a gel, which acts like a net, and an electric field the proteins can be separated, due to their negative charge. The amount of proteins in a proteome, even a comparatively small one, like the one of *E. coli* are very large such that one-dimensional electrophoresis does not sufficiently separate the proteins. As a solution the 2-d electrophoresis was developed, which on one-axis separates the protein mixture by isoelectric point and by size on the other axis [91]. Utilizing these two axes to separate the proteins increases the resolution of the assay [59]. Proteins are then quantified using any of the stains or antibody based methods. Due to the separation of the proteins, the stains or antibodies are no longer required to be specific.

Electrophoresis, however, also has drawbacks [197]. All proteins are denatured, meaning they lose their original structure, while specific types of proteins are lost all together. Additionally, the gels as well as other experimental steps like staining can introduce a lot of variability,

making reading the exact size or isoelectric point difficult. This in turn makes identification of specific proteins more difficult.

To alleviate this, electrophoresis can be coupled to *Mass-Spectrometry* (MS), which allows for a very specific identification of a protein [63, 106]. As it is not usually feasible to analyze whole proteins with MS, they are first split into smaller peptide chains, for example by enzymatic digestion. MS is usually coupled to a chromatography method, such as the one described above, to first separate the protein fragments by a physical-chemical characteristic. Afterward, the protein fragments are measured in the Mass-Spectrometer, a very common type of which is the *Time-of-Flight* (ToF)-Mass-Spectrometer. It is essentially a tube with a near vacuum, an ionization device at one end, and a detector at the other. The first step is the ionization of the protein fragments. As the ionization adds a charge to the peptide fragments, high-voltage gradients impart acceleration to the ions. They then fly through the vacuum tube and impact at the detector. The time of flight directly correlates with the mass-to-charge ratio, which correlates with the mass of the ion. In that way, all ionized peptides are detected and a spectrum is generated. The intensity and the recorded mass-to-charge ratio as well as the elution time of the chromatography allow for the quantification of the peptides and in turn the protein. In a final step the spectra are matched to references and the proteins are quantified.

While this describes the basic working principle of protein-quantification via ToF-MS, there have been substantial improvements and additions to the protocols. For one there are different types of mass analysis techniques and mass analyzers, like the combination of ion-traps and quadrupoles [170], each with their unique advantages and shortcomings.

Quantification via MS has several drawbacks, however [106]. Due to the fragmentation, the fragments can not necessarily be assigned to a single protein causing ambiguous signals. This gets more complicated the less well the protein mixture can be separated by chromatography. A major drawback of all proteomics techniques, especially compared to transcriptomics, is that rare protein species cannot be easily detected, as there is no amplification in any of the methods.

5.1.3 Metabolomics

As mentioned earlier, there is another class of molecules that in many cases have more direct impact on the actual phenotype: Metabolites. A big challenge in metabolomics is the heterogeneity of the researched metabolites [53]. So what are metabolites that they are this variable? In general, metabolites are the products of any step in the metabolism of an organism but normally excluding genetic information and proteins. So classical examples of metabolites include, but are by no means limited to: lipids, sugars and amino acids, in some organisms vitamins or antibiotics, are synthesized as well. As an additional stipulation, when talking about metabolites only small molecules, below a few Dalton, are considered. This then excludes the structured nucleic acids as well as larger proteins, which are covered by their respective omics anyway.

When investigating the metabolome the researchers need to be mindful of what part of the metabolome they want to investigate. Sometimes researchers do not have a specific set of metabolites that they want to detect or quantify, this is then called *untargeted metabolomics*. If a specific set of metabolites is targeted, for example, on a set of lipids it is conversely called *targeted metabolomics* and in this specific case often-times the term *lipidomics* is used.

Depending on which part of the metabolome is investigated, the techniques used to detect them vary quite drastically. While, similar to *proteomics*, MS is used frequently, other techniques like *Nuclear Magnetic Resonance* (NMR) can be used as well depending on the scenario [75]. However, chemically speaking, in contrast to proteins, metabolites are quite different from each other. And thus the methods historically used for the analysis vary widely as well.

But even when using MS the type of MS and the coupled chromatography method vary widely. The main reason for these differences is that the properties of the analyzed molecules cause different issues with the chromatography method. While lipids and carbohydrates can be detected using *Liquid Chromatography* (LC)-MS [44], it is not well suited for other compounds like very small, volatile compounds. For those a *Gas Chromatography* (GC)-MS is more suitable as it is especially useful for

small—that is smaller than 650 dalton [82]—molecules. For GC, however, the samples need to be either volatile themselves or to be derivatized to be volatile, which introduces yet another step in the experimental pipeline.

5.1.4 Multi-omics

This also highlights the most important fact that should be concluded from this background section. Each of the omics methods mentioned relies on a multitude of different and, at times, error-prone resources and steps. Starting with databases that need to be accessed for references to sample preparations, technical error rates and technique limitations, and even bugs in used software can be relevant. When visualizing such data and when interpreting the visualization, researchers and visualization designers should always keep in mind the uncertainty induced by the involved processes. And this also highlights why it might be important to combine the different types of *omics*-approaches into an integrated *multi-omics*-approach. There are overlaps in the type of information gained, as essentially all omics just try to answer: What is happening in this specific condition? But instead of being redundant, this information can be essential, to make sure that the measured changes in a specific omic actually are robust and not just an illusion spawned by the specific uncertainties of the utilized omic.

As mentioned, however, the current generation *omics* analyses generate giant amounts of data. This data is very connected in nature. As mentioned in beginning of this chapter all layers of omics-data, and with it their analysis, depend on and influence each other, which is reflected in databases that store data derived from omics experiments. Examples can be found in the *Reactome* [93] or *STRING* [265] databases which compile metabolic pathways and protein-protein-associations, respectively. Connectedness makes multi-omics data an ideal candidate for network analysis and network visualization.

5.1.5 Biological Network Visualization

This Section has been published in:

The following text, tables and figures was published as [35] N. Brich, N. Schacherer, M. Hoene, C. Weigert, R. Lehmann, and M. Krone. 2023. visMOP – A Visual Analytics Approach for Multi-omics Pathways. *Computer Graphics Forum* 42, 3 (June 2023), 259–270. <https://doi.org/10.1111/cgf.14828> only minor changes, those being the mentioning of a review paper ([70]) and the addition of a transition paragraph to the next two chapters, were made.

The visualization of networks can help to analyze the encoded relationships and elucidate underlying structures. It has been the target of research for a considerable time and has, thus, grown into an extensive field of research. Giving a comprehensive overview of network visualization is beyond the scope of this work. Therefore, we only give a very brief introduction and refer to state-of-the-art reports about the visualization of large [149], dynamic [19], multivariate [192], and multilayer [171] graphs, which summarize foundational methods and recent advances in the field of network visualization and provide pointers for further reading. Koutrouli et al. [143] recently gave an overview of commonly used graph visualizations for biological networks. An even more recent overview covering biological network visualization in was published by Ehlers et al. [70], with contributions of the author of this thesis. This review covers biological network visualization from a wide angle in order to give both an entry point to the subject and a comprehensive overview over the challenges of the field.

There are various ways networks can be displayed, depending on the use case. The most common representations are adjacency matrices and node-link diagrams [138]. While adjacency matrices make it easy to look up direct connections between nodes and—depending on the sorting—identify cliques and clusters, they are often considered to be less intuitive, and it is harder to find paths between nodes compared to a node-link diagram. Node-link diagrams, on the other hand, highly depend on

the layout, i.e., the spatial positioning of the nodes. A recent study [3] investigated and compared the performance of adjacency matrices, bipartite layouts, and node-link diagrams for five network overview tasks.

Especially for large networks with many nodes and edges, a good layout algorithm is crucial to enable users to detect important features like clusters, symmetries, and interconnected areas [143]. Many approaches to generate layouts have been proposed. The most simple layouts place the nodes on grids or circles. Many layout algorithms are based on simple simulations, e.g., modeling networks as a set of bodies connected by spring-like forces. While these approaches can be useful for analyzing smaller networks, they are often not useful for analyzing larger, more complex networks because they usually do not consider any property of the network.

Many disciplines in biology use networks and their visualization, most commonly node-link diagrams, to model and analyze relations between different entities e.g., phylogenetic trees showing the evolutionary relatedness of species [103, 217], or graphs modeling the connections between neurons in neurobiology [210, 302]. Molecular biology has a plethora of different application cases for networks, e.g., gene-regulatory networks [67, 133], protein-protein-interaction graphs [189, 259], or gene-variation graphs [72, 101]. The modeling of metabolic pathways is an important application area of networks in biochemistry [93, 134, 195]. These metabolic pathways describe how cells and, in turn, organisms function on a molecular level. They are, thus, instrumental in understanding any biological process. The nodes of the graph usually represent reactants that are connected by edges indicating reactions. Reactants can be a wide variety of biologically relevant molecules, e.g., proteins, small molecule metabolites, or DNA/RNA. As many organisms feature tens of thousands of biologically active entities (genes, proteins, RNA), and even more different types of interactions, the resulting networks are very large and complex. This makes it challenging to generate clear, meaningful visualizations of these networks. Thus, in the applied biomedical domain, many of the most well-known and widely used metabolic pathway visualizations are manually curated and laid

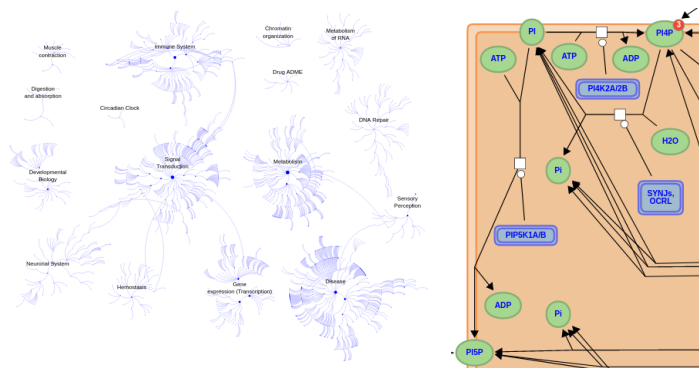


Figure 20: Metabolic pathways visualized by Reactome [93]. Left: excerpt of the overview of the metabolic pathway network where each metabolic region is the root for a hierarchy visualized using a radial layout. Right: excerpt of a detail reaction pathway containing molecule-level information. These pathways are located in nodes closer to the leaves of the radial hierarchy.

out diagrams. Recently, a first approach for the automatic generation of metabolic pathway maps that can retain many of the desirable features of the manually laid out diagrams has been presented [294]. However, it just considers information about the known interactions and does not incorporate omics measurements, making it unfeasible for our intended use case.

Reactome [93] is a pathway database offering information about molecular entities in conjunction with metabolic reaction pathways. The pathways are arranged in a hierarchy: each pathway is part of a multi-level hierarchy in which general metabolic topics constitute the roots and fine-grained reaction sequences constitute the leaves. The more complex pathway diagrams found at the roots and the levels further removed from the leaves are created manually (Figure 20). Reactome is part of the *ELIXIR* [56] initiative and offers its data and software under a Creative Commons license. This includes the instructions for all pathway diagram drawings, allowing a fast and efficient way of querying large amounts of data.

The Kyoto Encyclopedia of Genes and Genomes (KEGG) [130] is another widely used database offering information about different molecular entities and metabolic pathways. It presents the pathways in a flat hierarchy in which they are linked by functional connections. A separate class hierarchy groups the pathways. Similar to Reactome, the pathway diagrams are created manually to consider previous knowledge and conventions for a given process the diagram describes. However, it introduced a subscription model in 2011 due to funding issues and detailed information about how to draw the pathway diagrams is not available, making it less convenient to use.

Especially among practitioners, the two “*Biochemical Pathways*” posters by Roche [175] are popular. They show a comprehensive summary of known pathways and processes. The manually created graphs are structured into labeled sections on different high-level topics, e.g., *Citrate and Glyoxalate Cycle* or *Steroid Metabolism*. The lower level shows the actual chemical reactions with all participating molecules. Roche offers print and zoomable online versions of the posters, however, the data is proprietary and not publicly available, severely limiting the use for visual analytics applications.

The following two chapters each describe a novel approach to analyze multi-omics data utilizing network visualization. The first approach maps data of the three types of omic data previously discussed in this chapter, *transcriptomics*, *proteomics* and *metabolomics* to metabolic pathways. Using this mapping users can then find clusters of pathways that show similar patterns of regulation. The second approach uses a custom network-layout to analyze the protein-protein-associations found in the *STRING* database based on proteins-of-interest.

5.2 visMOP – A Visual Analytics Approach for Multi-omics Pathways

This Section contains previously published text

The following text, tables, and figures was published as [35] N. Brich, N. Schacherer, M. Hoene, C. Weigert, R. Lehmann, and M. Krone. 2023. visMOP – A Visual Analytics Approach for Multi-omics Pathways. *Computer Graphics Forum* 42, 3 (June 2023), 259–270. <https://doi.org/10.1111/cgf.14828>. Only minor changes were made. These are the fix of minor textual errors and removal of the abstract, redundant introduction, the funding information as well as the inclusion of the material previously placed in the supplementary material. The required background has been moved to Section 5.1.

5 *VisMOP* is an approach that allows for exploratory analysis of multi-omics datasets. To facilitate the analysis, we propose to use a clustered graph to visualize multi-omics datasets on a pathway level while representing the pathways as glyphs encoding the contained data. Our contributions can be summarized as follows. We present an approach that works with a variable number of different omics measurements, which are mapped against all known metabolic pathways from the Reactome [93] database. The resulting pathways containing the measurement data are summarized in an overview graph, which is visualized as a node-link diagram showing the interconnections between the pathways. For this network of pathways, we present a data-driven layout algorithm that can be customized by the users according to their specific needs, interests, and research questions. The pathways, which are the nodes of the overview graph, are represented by glyphs showing the omics measurements mapped to those pathways. That is, our approach allows users to compare different pathways concerning the measured data and facilitates the detection of correlations or mismatches in the data that require further investigation. We developed visMOP (visualization of Multi-Omics Pathways), a web-based prototype implementing our approach that also provides level-of-detail functionality for the overview

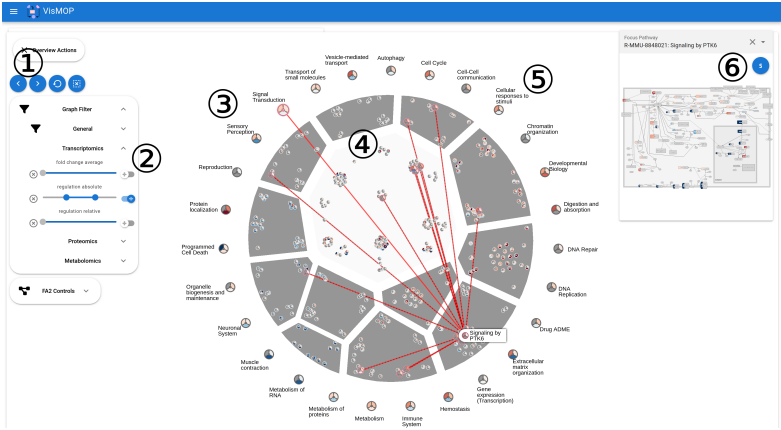


Figure 21: The main view of visMOP (visualization of Multi-Omics Pathways) showing the overview of the cluster graph with the Reactome superpathways laid out as a circle around it ③. The gray regions show clusters of pathways containing similar data, while the lighter region shows pathways not clustered with other pathways ④. A selected pathway is highlighted, showing both hierarchical as well as non-hierarchical connections to other pathways. ⑤ the detail view window is shown for a selected pathway. ② the visual filter options, as well as the option to control the intra-cluster layouts are displayed. The input data is shown in a collapsible table linked to the main overview visualization ①.

graph and multiple linked views for a more detailed inspection of individual pathways. We tested our approach on a real-life dataset together with domain experts.

5.2.1 Related Work: Multi-omics Pathway Analysis Tools

As mentioned in Subsection 5.1.5 there are several widely used and well known resources visualizing metabolic networks. In contrast to KEGG and the Roche biochemical pathway map, Reactome [93] offers an interactive visual analysis tool, which allows the user to map multi-omics measurement data onto the Reactome pathways. The visualization is rather simple: all omics measurements are aggregated and the pathway nodes, which are arranged in a hierarchical node-link diagram, are colored by either coverage or significance in coverage, or (when measurement data is supplied) the average overall supplied omics. It does

not influence the graph layout. Additionally, the data can be shown as “*Reactfoam*”, a hierarchical Voronoi treemap used to show the data in lieu of the node-link diagram. Any further analysis then has to be performed on a detail pathway level.

Paintomics [158] is a tool tailored to multi-omics. It can map different omics types—including, but not limited to: transcriptomics, metabolomics, proteomics, and different regulatory and region-based omics—to pathway maps from KEGG, Reactome, and MapMan [243]. Using these data, significantly enriched pathways are searched in these databases. The user can then select the pathways, and the molecular entities in the pathway visualization are then colored according to the corresponding data value. That is, the provided overview network visualization is mainly an entry point for the analysis of individual pathways. Our approach, in contrast, aims at generating a rich overview using layout and clustering techniques to generate insights from this overview by combining multiple omics, e.g., allowing users to identify groups of similarly regulated pathways and mismatches between the different omics levels. Pathview Web [161] follows a similar approach: it simply maps colors showing the regulation to the KEGG pathway drawings.

OmicsNET [306] leverages the third dimension to accommodate the complexity of multi-omics networks. Besides the omics data, it also allows the user to load additional information like SNPs for genomics or taxon information for studies involving multiple organisms. 3D visualizations, however, have not been shown to yield additional insights without the use of stereo-vision, i.e., VR or AR [171]. Such systems are typically not widely available in molecular biological laboratories, which are our target audience. Additionally, 3D network visualizations suffer from occlusion, perspective distortion, and usually require more user interaction [184]. To avoid this, we settled for 2D visualization. Furthermore, instead of focusing on details brought in by additional data, we want to show large-scale, pathway-overarching effects on the metabolism.

There are also frameworks targeted at systems biological network visualizations like VANTED [128, 228], which can accommodate multiple

omic types. While VANTED can also provide overviews, its focus lies on analyzing and visualizing biological data on a detail level in the context of systems biology, tracking changes in biological reactions and single molecule types like metabolites.

5.2.2 Tasks and Requirements

Our domain experts currently use different tools, such as g:Profiler [221], to either perform pathway enrichment analyses for each type of omics data individually or for subsets of features that overlap in the individual statistical analyzes of the different omics datasets. Similarly, MetaboAnalyst [202], which allows for an integrated analysis of metabolomics and either transcriptomics or proteomics data, combines the data without allowing for an exploration of the contribution of individual features from the different omics sets. This obscures the structure of the data, which is particularly problematic given that both parallel and antiparallel responses of different omics types can be relevant, e.g., an enzyme in a pathway can be upregulated, causing a higher amount of its product, but a higher amount of its product could also decrease the amount of the enzyme in a regulatory feedback mechanism.

Thus, our explicit goal was to develop an approach to unify the analysis and visualization of multiple omic types. Our domain experts explained that they need an approach to give an exploratory overview of the data at hand, which can provide an entrance point for hypothesis generation and further detail investigations. When formulating the tasks together with the experts, we considered several task taxonomies as a reference [33, 185]. A concrete overview task is to identify high-level groupings of metabolic pathways to uncover sets of co-regulated or connected pathways. Detail investigations require an additional view of the more granular aspects of the data, requiring the framework to support various additional by offering detail-on-demand functionality. They, for example, need more details about a specific pathway on a molecular level, thus requiring a detailed pathway view, as found in the Reactome or KEGG pathway visualization. Additionally, users might

want to compare different pathways to assess if similar omic patterns appear within them.

We propose a combination of a layout algorithm tailored to multi-omics data and glyphs that show information about the pathways. We combine these with a level-of-detail approach to further increase the scalability and to declutter the visualization. Both the layout and the glyph design were created with several requirements in mind. The requirement analysis was done together with four domain experts working in biochemical and biomedical research. In a series of semi-structured interviews, we first collected information about the data, analysis challenges, and common tools and workflows. From this information, we inferred an initial set of requirements, which we then discussed with the experts to ensure their validity. During this phase, we also showed several iterations of an earlier prototype [147] to the experts and reiterated on the requirements. Finally, we converged on the following set of five requirements:

- R1** The visualization must be able to integrate multiple omics types and assess differences in the measured data.
- R2** The approach must be able to scale up/down when more/less omics types are available.
- R3** The representation must give equal importance to all the included omics types.
- R4** The visualization needs to provide a summary of the individual pathways that allows for a quick comparison while also providing the opportunity for detailed inspection.
- R5** The network layout should make use of established pathway layouts.

5.2.3 Visualization of Multi-omics Pathway Networks

Some of the mentioned visualizations, like the Roche or KEGG pathways, helped to establish mental maps for many metabolic processes. These visualizations, however, present information on a detailed level and are not suitable as an overview. In an earlier iteration of visMOP, we attempted to generate a global network incorporating all detail pathways

utilizing the established detail layouts of KEGG, resulting in a very large graph with thousands of nodes and tens of thousands of edges. Based on the domain expert feedback, however, we realized that this approach does not yield much useful insight, which prompted us to change the focus of the application to giving an overview of the data, with details-on-demand for individual pathways.

This left us with the challenge of uniting existing mental maps with a data-driven overview. The overview maps of KEGG or Roche, however, are not particularly well suited for overview tasks as they are high-resolution abstractions of the detail pathways, difficult to use on regular-sized screens. Reactome's more abstract overview, on the other hand, is based on a hierarchy, making it more suitable. However, it can not be easily adapted to a data-driven layout. We, thus, chose to implement a novel layout based on the Reactome hierarchy, which maintains the mental map of users familiar with the Reactome hierarchy. Additionally, we maintain the detail pathways' layout of Reactome. Consequently, users who perform a more detailed analysis using visMOP can build on their previous knowledge.

Our approach combines multiple views and uses a glyph metaphor to encode pathways, as recommended by Cruz et al. [57]. We use a data-driven layout approach to tailor the visualization to the underlying data. Users can interactively adjust the input data, the parameters for the data-driven layout, and the appearance of the visualization to facilitate the analysis.

Pathway Network

The multi-omics pathway overview graph (see Figure 21, ③) consists of nodes representing different metabolic pathways obtained from the Reactome database. Relationships between the nodes—i.e., edges—correspond to two categories: They are either hierarchical, as defined by Reactome, or non-hierarchical, logical connections directly inferred from known molecular interactions (e.g., “pathway A produces metabolite X which is then used in pathway B”). There are two different subtypes of pathways: *superpathways*, which represent broad metabolic topics

5 (e.g. Signal-Transduction or Innate Immunity) and are positioned at the root of the pathway hierarchy, and non-root “regular” *pathways*, which include molecular-level information and can be found either as leaves or closer towards the leaves of the hierarchy. To create a clear visualization, we apply two different layout algorithms to these two subsets of nodes (**R5/R1**). For the superpathway hierarchy level, we use a simple circular layout, since there are only a few superpathways. To make it easy to find a certain superpathway, the nodes are sorted clockwise in alphabetical order by their names. The labels showing these names can be placed outwards to ensure good readability (see Figure 21, ⑤). In the center of this circle of superpathway nodes, the nodes that are lower in the hierarchy are shown (see Figure 21, ④). This is the main pathway network, consisting of pathways that include the actual molecular interactions. Here, a more complex, data-driven layout is applied, which is described in Subsubsection A Data-Driven Layout Algorithm for Pathway Networks. It should be noted that not the complete hierarchy provided by Reactome is displayed. We limit our overview network to the first pathways (in root-to-leaf order) for which Reactome associates a molecule-level diagram (thus omitting intermediate levels between the superpathways and the actual molecular interactions).

As the network is densely connected, especially to the superpathway nodes on the surrounding circle, the edges would cause intense visual clutter if all of them were displayed. This would make the identification of source and target nodes and paths between them very difficult. To alleviate this, edges are hidden by default. Only when a node—superpathway or pathway—is hovered, the edges connected to this node are shown.

A Data-Driven Layout Algorithm for Pathway Networks

Many node-link visualizations utilize simple layout algorithms that only consider the graph topology to generate a somewhat understandable visualization. These layouts usually minimize overlaps and reduce edge crossings; however, since they usually do not consider non-topological information, i.e., the underlying data values, they rarely result in layouts

that are meaningful for the interpretation and analysis of the data, in the sense that the semantics of the data are taken into account. Therefore, we devised a layout algorithm that takes the characteristics of the underlying data into account.

Our proposed layout is based on the idea of using clusters, similar to the Grid- and Modularity-based Layout algorithm by He et al. [110]. For the clustering, the values and statistics about the supplied multi-omics data are taken into account, such as the average *fold change* per omics, the number of regulated entities per pathway, etc. (**R1**). The fold change is commonly used in bioinformatics to express the change of a given measurement value from an experimental condition—most often the control—to another one. Note that either all available parameters or only a user-defined subset can be used as input. While these metrics, available in the presented prototype, are not exhaustive and include others (see Table 2), they serve to highlight the user-customizable approach of selecting data metrics that are relevant to the analysis case. For further, more specific applications additional metrics tailored to the use case can be implemented straightforwardly.

Density-based clustering in high-dimensional Euclidean spaces can become problematic due to the sparseness of the space—a phenomenon often dubbed the “curse of dimensionality” [13, 116]. One way to overcome this is to use dimensionality reduction methods like UMAP [172] as a preprocessing step [13]. As an input to UMAP we use the parameters derived from the input data, which are normalized by transforming them to a Z-score. We apply UMAP twice: once, if more than two features are chosen, to generate a lower-dimensional space (with a default $2/\sqrt{N}$ -target dimensions) for clustering with OPTICS [10]. For visualization purposes time we project the data a second time, this time to two dimensions, to determine cluster centers as the basis for a next step. We chose OPTICS, because it is an unsupervised method that determines the ideal number of clusters and can classify entities as noise if they do not fit into a cluster. We run the clustering several times with different values for the minimum size of a cluster to find an ideal clustering by means of the silhouette value [230] for differently-sized datasets. That is, the user does

Table 2: Selectable features for dimensionality reduction and clustering derived from the input data

Transcriptomics: Number of values	Absolute number of measured Transcriptomics values mapped to the pathway
Proteomics: Number of values	Absolute number of measured Proteomics values mapped to the pathway
Metabolomics: Number of values	Absolute number of measured Metabolomics values mapped to the pathway
Transcriptomics: % regulated	Percentage of measured Transcriptomics values mapped to the pathway which are considered regulated (over and under user-defined upper-limit/lower-limit)
Proteomics: % regulated	Percentage of measured Proteomics values mapped to the pathway which are considered regulated (over and under user-defined upper-limit/lower-limit)
Metabolomics: % regulated	Percentage of measured Metabolomics values mapped to the pathway which are considered regulated (over and under user-defined upper-limit/lower-limit)
Transcriptomics: % unregulated	Percentage of measured Transcriptomics values mapped to the pathway which are considered unregulated (under and over user-defined upper-limit/lower-limit)
Proteomics: % unregulated	Percentage of measured Proteomics values mapped to the pathway which are considered unregulated (under and over user-defined upper-limit/lower-limit)
Metabolomics: % unregulated	Percentage of measured Metabolomics values mapped to the pathway which are considered unregulated (under and over user-defined upper-limit/lower-limit)
Transcriptomics: % with measured value	Percentage of pathways Transcriptomics values with measured value
Proteomics: % with measured value	Percentage of pathways Proteomics values with measured value
Metabolomics: % with measured value	Percentage of pathways Metabolomics values with measured value
Transcriptomics: Mean expression above limit	mean of transcriptomics measurement level from the values mapped to the pathway above the user defined upper-limit
Proteomics: Mean expression above limit	mean of Proteomics measurement level from the values mapped to the pathway above the user defined upper-limit
Metabolomics: Mean expression above limit	mean of Metabolomics measurement level from the values mapped to the pathway above the user defined upper-limit
Transcriptomics: % values above limit	percentage of Transcriptomics measurements mapped to the pathway above the user defined upper-limit
Proteomics: % values above limit	percentage of Proteomics measurements mapped to the pathway above the user defined upper-limit
Metabolomics: % values above limit	percentage of Metabolomics measurements mapped to the pathway above the user defined upper-limit
Transcriptomics: Mean expression below limit	mean of Transcriptomics measurement level from the values mapped to the pathway below the user defined lower-limit
Proteomics: Mean expression below limit	mean of Proteomics measurement level from the values mapped to the pathway below the user defined lower-limit
Metabolomics: Mean expression below limit	mean of Metabolomics measurement level from the values mapped to the pathway below the user defined lower-limit
Transcriptomics: % values below limit	percentage of Transcriptomics measurements mapped to the pathway below the user defined lower-limit
Proteomics: % values below limit	percentage of Proteomics measurements mapped to the pathway below the user defined lower-limit
Metabolomics: % values below limit	percentage of Metabolomics measurements mapped to the pathway below the user defined lower-limit
Not omic-specific: % values measured over all omics	Percentage of all omics measured and mapped to the pathway combined in relation to all omics entries in the pathway.

not have to parameterize the OPTICS clustering (for epsilon, we use the default value of ∞). We assign the noise entities to a virtual cluster that is marked in the final visualization to illustrate that these entities are not similar (region with light gray background; see, e.g., Figure 21, ④).

In these next steps, the space formed by the above-mentioned high-level hierarchy nodes is partitioned in a way that both the distance from one cluster to another and the proportion of nodes contained vs. space allocated are considered. He et al. [110] use rectangular regions for the clusters in this step and try to optimize their placement to reduce edge crossings. In contrast, we use a circular layout, in which a Voronoi treemap is used to allocate space [17, 193]. This results in a layout in which similar clusters are placed next to each other (see Figure 22).

After the cluster space is allocated, the nodes are distributed inside their cluster regions. For this final step in our layout pipeline, we use an adapted version of the ForceAtlas 2 [123] layout algorithm, modified to restrict the layout to a convex polygon. We randomly position, the pathway nodes inside the Voronoi-cell, disconnecting it from the UMAP embedding space. We apply weights using a simple weight function. Nodes belonging to the same Reactome superpathway will be weighed more than other edges (**R5**). Thus, all pathways in a cluster that belong to the same superpathway will be placed closely together. This shows the semantic relations of the pathways based on the Reactome hierarchy and, thus, facilitates the analysis using biological domain knowledge. He et al. [110], in contrast, again only optimize the layout by using intra- and inter-cluster connectivity and minimizing edge crossings.

Pathway Glyphs

Since the nodes of the network represent whole pathways and multi-omics data is used, the amount of data per node can be large and diverse. This makes it challenging to give the user an overview of the measured data per pathway. We propose a radial glyph representing the multi-omics data contained in the respective pathway. Our glyph is divided into equally-sized radial sectors, depending on the number of omics to be displayed, i.e., two half circles for two omics types, three third-circles

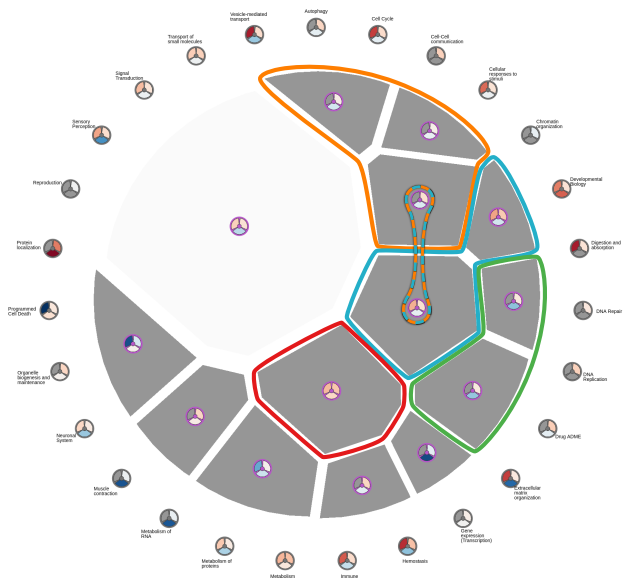


Figure 22: visMOP overview annotated to show similar cluster aggregate nodes in groups. As the aggregated nodes indicate the average regulation of the supplied omics data of the pathways contained in the cluster, this grouping thus also groups by average regulation. The cluster regions similar to each other are placed in proximity to each other, leading to contiguous groupings. The —-dashed line highlights the fact that clusters, although sharing similar coloring for two of the three omics, can be separate due to the lack of measurements in the third omic type. The other colors indicate different trends in the respective clusters.

for three omics types, etc. (**R1/R2**). Each sector is split into an outer and inner ring, as shown in Figure 23 (right). The outer ring presents the regulation data per measured entity and is, therefore, again divided into equally sized radial sub-sectors. Each sub-sector represents one omics measurement for an entity in the pathway (e.g., the transcription rate of a certain gene or the abundance of a certain protein). To convey this information in a simple way, we color the sub-segments according to their sorted measurement data, using a diverging color map from red (high positive regulation value) over white (neutral, no up-/down-regulation) to blue (high negative regulation value) (**R1**). The inner ring

is used to display the number of measured molecular entities in relation to the absolute number of entities of the same type that are contained in the given pathway (dark gray portion). An annotated version of the glyph additionally shows these key information as text (Figure 29).

We also designed a simplified version of this glyph for scenarios in which the zoom level makes the detailed reading of values infeasible. The simplified glyph is shown in Figure 23 (left). Here, the average of the regulation values for each omics type is computed and mapped to the corresponding color. Furthermore, we do not show the inner ring, thus providing more space for showing the average regulation value. This supports the *at-a-glance* analysis of the average regulation of the respective omics level per pathway (**R4**).

We chose a radial glyph for several reasons. Radial glyphs have been successfully applied in a similar scenario where a large amount of glyphs is drawn in proximity to each other [85]. As nodes can be positioned rather close to each other in our layout, a linear glyph could lead to that undesired effect that bars could appear to be continuations of other, unrelated glyphs. Additionally, linear glyph designs could imply different degrees of importance to the different omics types due to the sorting. That is, radial glyphs allow for a clear separation of single pathways and do not impart any inherent hierarchy to the different omics (**R3**). Our circular design also easily scales with the number of displayed omics types (**R2**).

5.2.4 Prototypical Application Design and Implementation Details

We implemented our approach as a prototypical web-based client-server application shown in Figure 21. On the server side, we used a python flask [229] backend to perform the mapping of the measurement variables to the Reactome pathways and to calculate the clustering using the UMAP [172] and OPTICS [10] implementations of scikit-learn [207]. Furthermore, all Reactome data like the detail pathways are served by the flask backend.

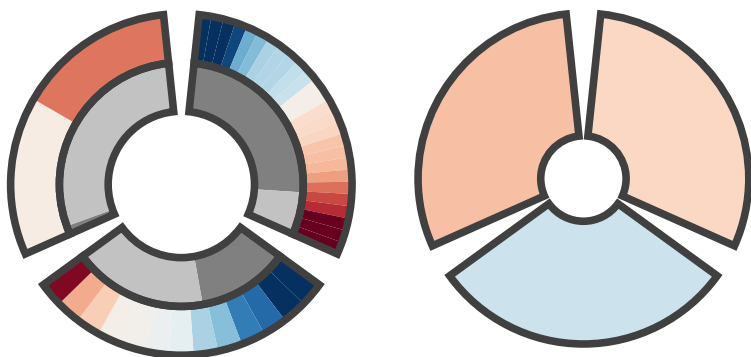


Figure 23: Glyphs for different levels of detail using a blue-white-red diverging color scale () to encode measurement values. The high-detail version shows discrete circle segments for each entity and the measurement coverage, while the low level-of-detail version uses averaging to reduce the amount of information.

The remaining functionality of the prototype is facilitated by the client-side frontend, developed using TypeScript and Vue.js. For the network visualization, d3.js [29], a Voronoi treemap plugin for d3 [151], and customized versions of the WebGL-accelerated Sigma.js [215] and ForceAtlas2 [123] are used. Our prototype contains four basic UI sections: an input sidebar for data selection, the main canvas showing the overview pathway network, a resizable view displaying the detail pathway, and a menu to parameterize the filtering and control the intra-cluster layout. For the detailed algorithmic pipeline see Figure 24.

Since the visMOP network is designed to support the user in examining different data sets for different characteristics, there are various ways to customize the network to suit personal needs. The users can filter the input data by any data column supplied with the data. This can be but is not limited to, the data variable used for mapping onto the Reactome hierarchy. In addition, the user can select from a list of statistical parameters per omics type, according to which the pathways should be clustered, and can, thus, influence the layout. After these two settings have been made by the user, the data is sent to the server to

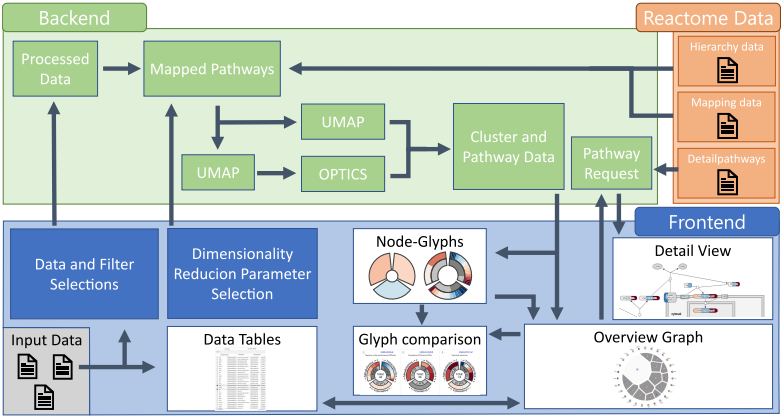


Figure 24: visMOP pipeline overview. Using the interface of the presented prototype, users can input up to three different omics types (transcriptomics, proteomics, metabolomics). These input data is displayed in interactive tables. The user can set up filtering for the data and can choose variables derived from the data to be used in dimensionality reduction. The data is then sent to a backend to be processed for mapping to the Reactome pathways. Using the selected parameters, the pathways are embedded into a lower dimensional space with UMAP. With OPTICS, clusters are detected in this lower dimensional space. Using UMAP a second time, a 2D embedding is generated from the mapped pathway data. Cluster data, including the mapping, the Cluster assignment and cluster positions (as defined by a centroid) are sent to the front end for visualization. The mapping data is then used to generated glyphs for each pathway. These glyphs are then used as nodes in the overview graph. Utilizing the cluster assignment and cluster centroids, a hierarchical treemap is generated. Interaction with the overview graph, coupled with linkage to other visualizations, offers the user an exploratory analysis of the supplied data. The other visualizations include a detail view, in which specific pathways are queried, and the mapped data is visualized in the Reactome pathway layout. Furthermore, multiple pathways can be selected for comparison, here the glyphs become interactive and individual measurements and omics can be investigated.

be summarized into pathway datasets. These are then sent back to the client together with the clustering data and subsequently, the network is drawn.

Depending on the zoom level, the pathway glyphs are either shown in full or reduced detail, as explained in Subsubsection Pathway Glyphs. When the user further reduces the zoom level, the pathway nodes get

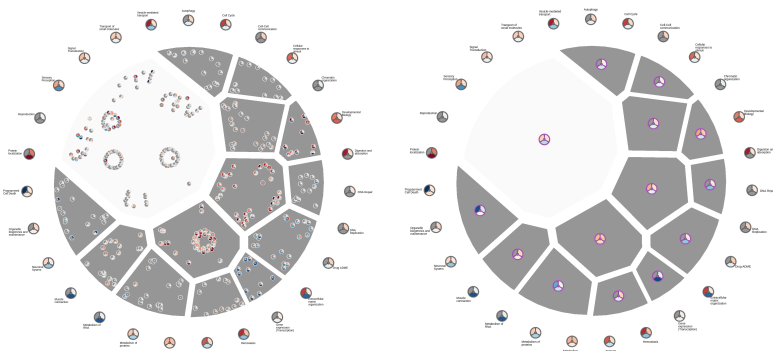


Figure 25: When the zoom level is reduced below a threshold, the pathway nodes inside the clusters are aggregated into larger nodes utilizing the low level-of-detail glyph.

aggregated into one node per cluster (Figure 25). Each of these cluster nodes is shown using the low-detail variant of the proposed glyph design to show averages over all pathways that are part of the aggregated cluster. If the user selects a node, a focused and zoomed-in state of the respective cluster is shown, which again contains all pathway glyphs of this cluster (Figure 26) (**R4**), i.e., hiding all other clusters.

As the size of the glyphs only allows for a limited in-detail inspection, the user can select one or more pathways to add them to a small multiples view, the pathway compare functionality, which allows the user to compare different pathways (**R1/R4**). As shown in Figure 31, the interactive glyphs additionally contain text labels showing the data values. This includes the total amount of entities, the proportion of measured entities, as well as the average measurements per omic when the corresponding section is hovered. The user can interact with these glyphs, switching their order via drag-and-drop or via mouse-over, which will display the name and associated data value of the respective molecular entity, or the average value of the respective omics category. Furthermore, utilizing the input data tables, the user can interactively search and select one or multiple molecular entities, highlighting pathways containing the union

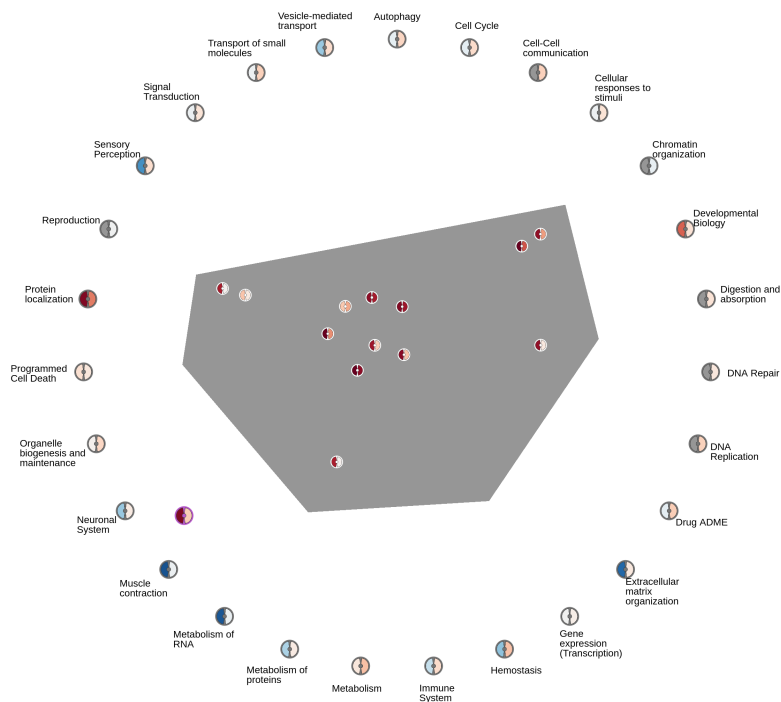


Figure 26: Clicking a cluster aggregate node hides all structures not belonging to the cluster and instead displays the cluster in a zoomed state in the center of the canvas.

and intersection sets of the selected entities, respectively (Figure 27). The table (Figure 21, ①) is located in a collapsible section on the left of the application, shared with the pathway compare functionality, between which it is possible to switch.

To reduce clutter and increase the focus on the selected characteristics, the user can apply filters, which act on the pathway level (see Figure 21, ②). Here, several predefined filter criteria can be used to hide pathways in the visualization. In our prototype, these criteria are: belonging to a Reactome pathway superpathway, sum of measured values (relative and absolute), average changes, and relative or absolute coverage of the pathways for each omics type.

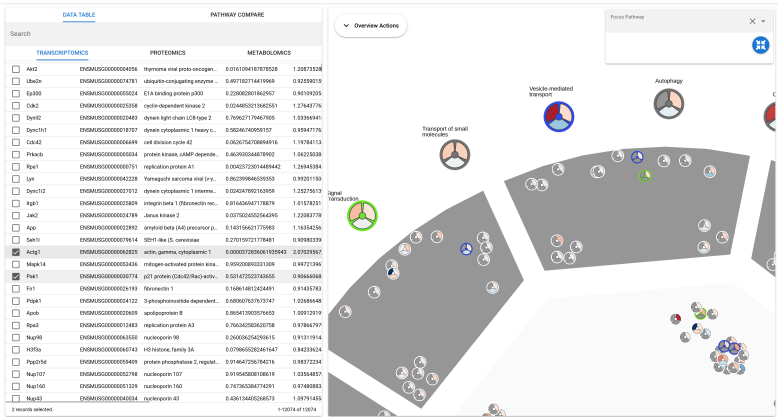


Figure 27: Selecting entries in the input data table highlights pathways containing the selected entities in blue, indicating a union, or green, indicating an intersection of all selected entries.

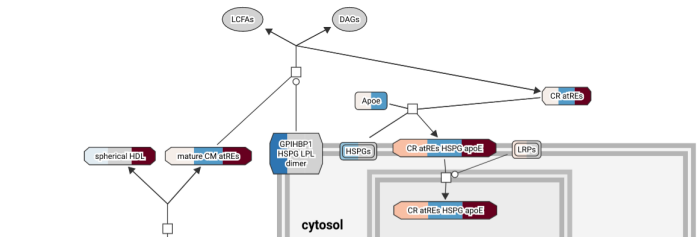


Figure 28: Excerpt of a detail visualization of a pathway, using the Reactome layout. Measured entities are mapped onto their corresponding nodes. Metabolites are colored by the corresponding measurement value. Single protein nodes are split at the center, showing color information about transcriptomic and proteomic regulation. Complexes are divided into thirds and colored by transcriptomic, proteomic, and metabolomic regulation. These complex nodes can be clicked to show a labeled detail glyph containing all the members of the complex (analogous to the pathways, as shown in Figure 29).

The overview visualization constitutes the centerpiece of this visual analytics approach. However, it has—by design—certain limitations with respect to the details that are available when zooming in. Due to the missing details about the interactions of the molecular entities, it can be

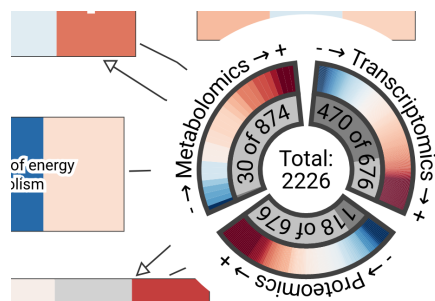


Figure 29: Using the detail view of the “Metabolism” super pathway, an Overview glyph of the *Metabolism of lipids* sub-pathway can be shown by clicking on the corresponding pathway node.

difficult to investigate more complicated research questions or to further refine hypotheses generated via the overview visualization. Thus, users can select a pathway to display the pathway diagram on a molecular level (see Figure 21, ⑥). We choose the layout and node/link encoding provided by the Reactome pathway browser (see Figure 20 right) to preserve mental maps established by users familiar with Reactome (R5). We draw the diagram in grayscale and apply the same diverting red-white-blue color map that was used for the regulation values in the pathway glyphs to the molecular entities in the pathway diagram (see Figure 28). That is, the omics measurements are color-coded onto the corresponding visual elements, which was shown to be beneficial in other tools [158, 161]. This way, users can investigate the regulation of the molecular entities in detail and immediately see where no data values are available. The pathway diagrams can also be panned and zoomed.

A central element of the Reactome detail pathway diagrams are complexes. This is due to the fact that many biological processes are not facilitated by single molecular entities, but instead through an interplay of several molecules or multi-mers of the same one. To keep the familiar appearance of the original Reactome visualization and to reduce visual clutter, the complex elements are only colored by the average value per entity type (like the low-detail glyphs). Analogous to the overview

glyphs, the user can click on a complex to show a glyph that shows each of the molecular entities partaking in the complex (Figure 29).

5.2.5 Results and Discussion

In this section, we present an exemplary workflow using real-world multi-omics measurements, which was executed together with a domain expert, and the feedback provided by the expert. To this end, datasets were supplied by the expert that had previously been analyzed separately for the three different omics domains.

Our visMOP approach primarily focuses on generating a rich overview to facilitate an exploratory data analysis, while allowing more detailed observations using several linked detail-on-demand views and techniques. To our knowledge, none of the other methods and available tools offers a data-driven layout approach that is interactively parameterizable and allows expert users to generate network overviews that allow them to focus on exploring the data to discover entry points for further analysis. The recomputation of the layout only takes in the order of seconds for typical data set sizes. Consequently, users can adapt the layout interactively while investigating the data, in case the user wants to explore the data with a different focus.

Below, we show the application of our prototypical implementation of visMOP on a real-life experimental dataset from *M. musculus* (house mouse) by performing a basic workflow to generate new hypotheses. The data [118] were acquired in an experimental study investigating the effects of high-caloric feeding on mitochondrial lipid metabolism in the context of fatty liver disease. Samples were taken from liver tissue, and transcriptomic analysis was performed on whole tissue. Proteomics was performed on isolated mitochondria. Lipidomics analysis, i.e., a metabolomics analysis focused on lipids and lipid metabolites, was performed on both whole tissue and isolated mitochondria. This results in measurements for a reference control group and an interventional group, the group that was fed the high caloric feed. From these, differences per measurement, the fold change, can be calculated. From more than 12,000 and 900 measured transcripts and proteins, respectively, 5,936 and

654 entities could be mapped to Reactome identifiers. The lipidomics measurements were mapped to 37 identifiers, corresponding to either individual lipids or sums of lipid classes with mitochondrial or pan-cellular localization. While this represents a considerable loss of data, Reactome takes great care to only include interactions backed by literature evidence, increasing confidence in the overview [127, 277]. As new evidence is published, new interactions are added to Reactome with each new revision, reducing the loss of data.

For the initial clustering and layout, no input parameters were changed in visMOP to fully utilize the data, i.e., no filtering or setting of the metrics used for clustering. The first inspection of the overview revealed that it was not focused enough for the planned investigation. Therefore, the input values that are considered in the clustering step of our layout pipeline were adapted: the minimum fold change limit for the metabolites was removed (since small changes in lipid concentration were of interest), unregulated entities were excluded from the clustering (since they are of little relevance here), and size of the Reactome pathways was not considered (since we did not want to focus on pathway size). After this adaption of the layout, the complete overview network was again inspected to search for any noticeable high-level trends (Figure 22). In this case, a first general observation is that several clusters contain glyphs with a rather subdued coloring (Figure 22 ) indicating relatively little change in the omics. While other clusters show strong tendencies to red (Figure 22 ) indicating up-regulation, or mixed between red and blue (Figure 22 ) or blue and neutral (Figure 22 ). Another interesting observation is the similarity in transcriptome and proteome coloring of two cluster aggregate nodes (Figure 22 -dashed), however, there is no metabolic data associated to the pathways in one of the clusters, which is a reason why they are separated. Furthermore, it is of notice that the aggregated cluster nodes in Figure 22  appear relatively similar. As the layout, however, uses data statics other than the average fold change, clusters might form that do have similar average fold change characteristics but differ otherwise.

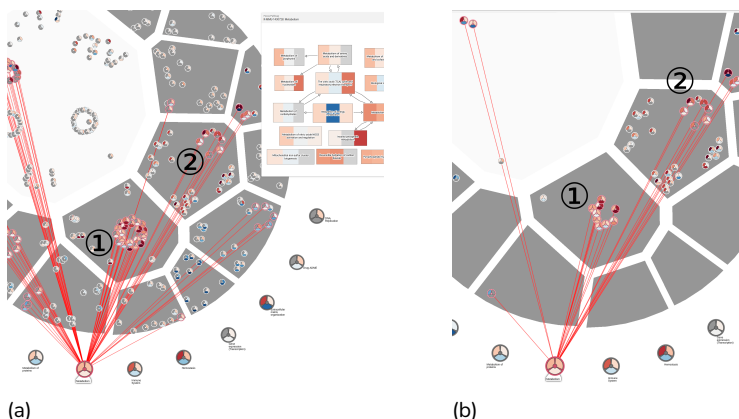


Figure 30: (a) visMOP overview with the “Metabolism” superpathway selected, highlighting all pathways in the respective hierarchy, and showing the detailed view of the “Metabolism” superpathway. (b) The visual filtering utility of the prototype was used to filter pathways not containing metabolites

As the data stem from an experiment designed to investigate the response to an energy-rich diet, an obvious starting point would be to inspect pathways that are a part of the “metabolism” superpathway (Figure 30). To help find agglomerations and patterns in these pathways, the intra-cluster layout positions pathways belonging to the same superpathway in proximity to each other and the superpathway glyph. Selecting the latter highlights all pathways assigned to, in this case, the metabolomics superpathway. An immediate observation is that many of the associated pathways are located in the two clusters containing mostly red, i.e., up-regulated, pathways.

In agreement with the predominantly red color in the zoomed-out bird’s eye view, the details of the metabolism superpathway glyph reveal that most of the transcripts, mitochondrial proteins and lipid metabolites detected and assigned to the pathway are increased in the liver of mice fed the energy-rich diet (Figure 29). The inner sections of the glyph also show that the coverage is much higher for the transcripts than for proteins and metabolites, in agreement with the fact that these datasets

were obtained from isolated mitochondria and using lipidomics, i.e., a lipid-focused metabolomics analysis.

Next, the individual clusters connected to the *Metabolism* superpathway can be inspected. The bird's eye view shows that pathways pertaining to the metabolism superpathway form sub-clusters in multiple clusters and in the “noise”-cluster (Figure 30). Among these sub-clusters, several do not contain data from all three omics types, in agreement with the preselection due to the mitochondria- and lipid-focused protein and metabolite analyses—a fact that can be easily visualized using the filter functionality (Figure 30b).

One particularly large sub-cluster was selected for a detailed assessment of the individual pathways using the “compare pathways” functionality. Figure 31 shows the pathways contained in this sub-cluster that are directly related to lipid metabolism. This view allows for a more detailed interpretation at the pathway level. In this example, one striking observation is that not only metabolites—i.e., lipids—but also transcripts and proteins are generally upregulated in pathways related to lipid metabolism. This includes several pathways concerned with the metabolism of fatty acids, in agreement with the original study [118] and indicates that the increased amount of fatty acids supplied with the fat-rich diet in the study caused an increased utilization of these in the abovementioned pathways. In contrast, individual proteins and transcripts assigned to the fatty acyl-CoA biosynthesis pathway, i.e., a pathway concerned with the synthesis of fatty acids, showed a higher degree of downregulation. The metabolites associated with this pathway are colored in red, indicating an increase, but clicking this section of the icon quickly solves the apparent contradiction and reveals that these are fatty acids, likely derived from the diet and not from synthesis in the organism.

This concludes the exemplary workflow on how a hypothesis can be generated using visMOP, which can then later be confirmed via different means. In this case, we showed a tendency in the data which was previously analyzed in detail and published [118]. There are a multitude of different entry points for further detail analysis, e.g., other clusters focusing on other processes. Additional analyses of the same dataset could,

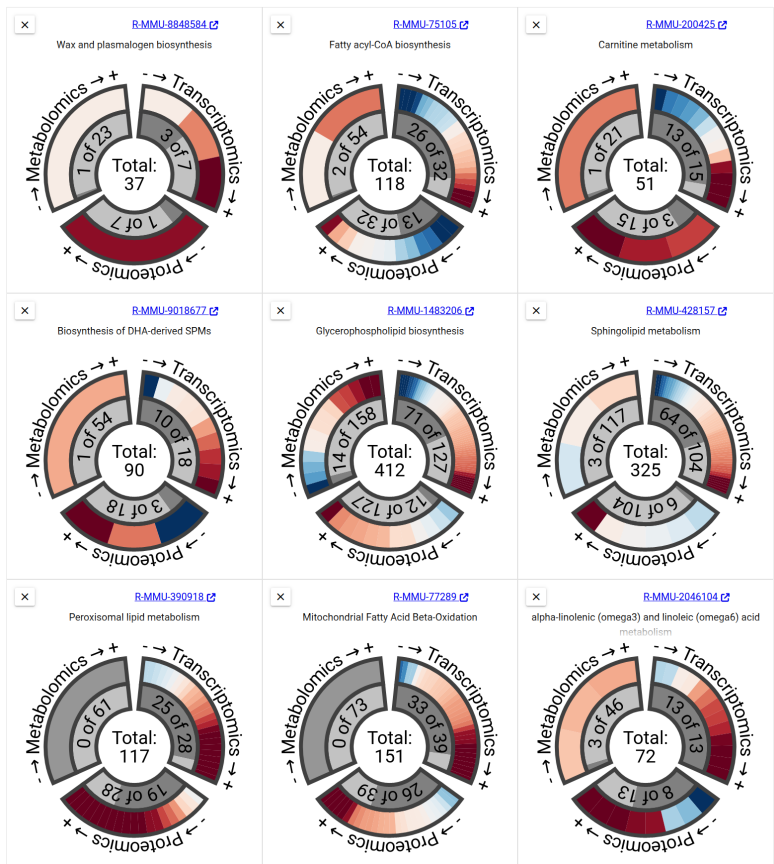


Figure 31: By selecting pathways for the “Pathway Compare” utility of our prototype, high detail versions of the pathway glyphs can be easily compared. Here, pathways associated with lipid metabolism are shown for Cluster 1 (Figure 30 ⑤)

for example, investigate the makeup of the blue clusters, indicating downregulation, or the makeup of the with respect to the different omics differentially regulated cluster. This shows that the presented overview approach can be successfully used to open up the dataset to different research directions, increasing the knowledge yield of time-consuming and costly experiments.

In general, the experts reported that our approach offers an innovative way of investigating the data. They particularly mentioned that the clustered view is helpful to get quick insights into the general structure of and an overview of the data. At the same time, they felt that visMOP's layout and design motivated them to explore the data.

5.2.6 Conclusion and Outlook

We presented visMOP, a visual analytics approach for multi-omics pathways that combines pathway-level glyphs with a data-driven layout algorithm to facilitate an exploratory analysis workflow. Our main contribution is the overview visualization, not only abstracting the data but using the underlying structure to support the exploratory analysis. In our prototypical implementation, users can investigate experimental data from up to three different omics domains, to explore the measured data and search for interesting trends, correlations, and outliers. However, our approach is not theoretically limited to the presented omics types and could be extended straightforwardly to support more types. Similarly, while we currently use pathway data from Reactome, extending our approach to utilize other databases like KEGG would be straightforward. Filtering, details-on-demand, as well as level-of-detail visualizations, allow the development of more focused hypotheses for further bioinformatics or biomedical analyses. We demonstrated the applicability of our approach in a basic workflow using a real-life data set from mice.

During the design process and the implementation of visMOP, several important insights were gained. In the beginning [147], we attempted to create a unified overview using existing pathway visualizations, like KEGG pathway maps, to show all details and fully maintain the user's mental map. This, however, resulted in a visualization that was too cluttered and difficult to use, as confirmed by our domain experts. We realized that combining familiar visualization, i.e., decorated detail pathways utilizing known layouts, with novel types of visualization yields a more accessible result. One such novel visualization is the data-driven clustered overview. We also realized that for complex and detailed data, a "fingerprint"-style [137] visualization, like our glyphs, can be more

helpful than a more fine-grained display of the data. This is especially true for the presented overview concept, where general trends in the data might be more relevant than specific results, and could be transferred to similar works in which giving an overview is the main focus.

Current Limitations and Future Work:

In its current form, visMOP only supports the direct comparison of one experimental group with a reference at one given time point. While this satisfies many experimental setups, some scenarios are not covered; including time-series experiments and multi-condition setups. These directions offer challenges, as the cluster space might change over time, making a consistent yet clutter-free visualization difficult.

While the glyph design allows an at-the-glance overview of the presented use cases, the radial division for the omic types might result in limited scalability. As the detail-level Reactome pathways themselves are limited in size, to be readily understandable by a human, this only affects the overview pathways. For the overview pathways, differentiating the entities can indeed become difficult, but in these cases, the outer channels act as a “fingerprint”, allowing for a qualitative comparison of the overview pathways. Furthermore, while this is not an issue for three omics types, higher amounts of omics would make memorizing the order of omics more cumbersome. However, as most biomolecular experiments do not collect more than the omics types presented, with maybe one or two more, this has little impact on the applicability of the glyph.

Both hierarchical and direct connections between the different pathways are only drawn on demand. This is due to the high amount of edge crossings and the generally high level of connectivity of the graph. While this preserves the readability of the graph, it also makes it hard to evaluate connectivity trends of the graph as a whole, which contradicts the idea of an integrated, comprehensive data analysis we try to foster with visMOP. Thus, we plan to employ techniques like edge bundling to reduce the cluttering introduced when displaying the entirety of the edges while simultaneously making trends with regard to the edge connections more visible.

To increase the visual analytics capabilities of visMOPs, pathway enrichment analysis could be added. These enrichment values could either be used as further input values for the overview layout, or filtering pathways. This could reduce the chance of including irrelevant pathways in the clustering or the visualization. Additionally, further aspects for analyzing the specific omics are conceivable. One example would be the integration of protein-protein interaction networks, which can be used to investigate if proteins of interest found during the previous analysis interact with each other.

While we already offer user choice by filtering the input data and by means of selecting from a variety of derived variables, further possibilities to influence the clustering would increase its value. Weighting the derived variables, e.g., using UMAP with a different distance metric, would allow users to emphasize some attributes without disregarding others. Thus, investigating the effect of different distance metrics and even dimensionality reduction and clustering approaches on the quality of the results would be an interesting direction for future research.

5.3 ProtEGOnist – Visual Analysis of Interactions in Small World Networks Using Ego graphs

ProtEGOnist is an approach originally developed to visualize protein-protein-associations from the *STRING* [265] database as a contribution to the Bio+MedVis Challenge 2023 [145]. *ProtEGOnist* won this challenge and as a follow-up we decided to expand the scope from protein-protein-associations alone, to be applicable to all kinds of *small-world networks*. This version, with its expanded scope, was published. Therefore, the primary example highlighted in the paper is not one of the biomedical, but instead from the social science domain.

This Section contains previously published text

The following text and figures was published as [34] N. Brich, T.A. Harbig, M. Witte Paz, K. Nieselt, and M. Krone. 2024. ProtEGOnist: Visual Analysis of Interactions in Small World Networks Using Ego-graphs. *Computer Graphics Forum* 43, 3 (June 2024), e15078. <https://doi.org/10.1111/cgf.15078>. Only minor changes were made. These are the removal of the funding information as well as the inclusion of the material previously placed in the supplementary material

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 [176, 291]. 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 [132, 190].

The small world property is also prevalent in many biological networks [18] like 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

5.3 • ProtEGOnist – Visual Analysis of Interactions in Small World Networks Using Ego graphs

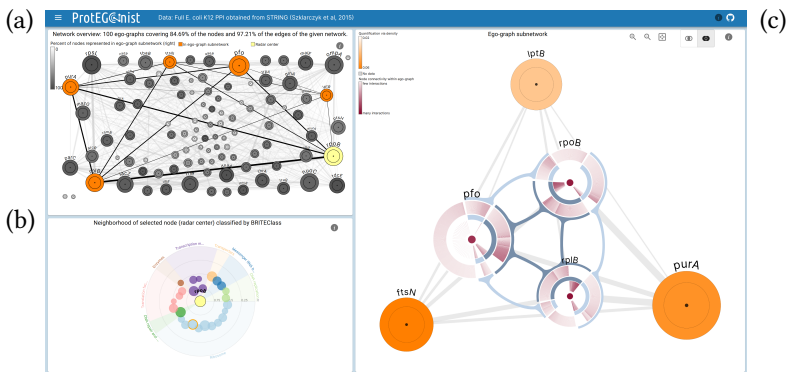


Figure 32: *ProtEGOnist* uses ego-graphs for the visualization of small-world networks (in this case, the *Escherichia coli* protein-protein interaction network). (a) The network of ego-graphs visualized as aggregated glyphs gives an overview of the network with edges visualizing similarity. The edges are weighted by the Jaccard index of the two respective ego-graphs represented by the nodes. (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.

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 [84]. 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 [194]. 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 [86] and their ability to find a job [208]. 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 [144]. Common path lengths for PPI networks are between four and five [298], thus, contacts with a distance higher than two tend to cover very large portions of the network [6].

To study such local subnetworks around nodes of interest, ego-graphs can be used [260]. 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 [305], i.e., “friends-of-friends” networks.

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 32). Initially, *ProtEGOnist* was submitted as a contribution to the Bio+MedVis Challenge 2023 [145] for redesigning the visualization of a specific PPI network by Gonçalves *et al.* [98]. The challenge dataset 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 dataset, *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 (Subsubsection Ego-graph Concept & Visualization Design). 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 (Subsubsection Glyph and Ego-Graph Group Redesign). 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 (Subsubsection Visual Interface & Application Design). Using the taxonomy proposed by Filipov *et al.* [84], 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.* [278], 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 [121], i.e., a social network (Subsubsection Co-author network). The other two show that our approach can be used for domain-specific datasets, such as visualizing PPI networks: we applied *ProtEGOnist* to a PPI network of *Escherichia coli* (Subsubsection *lac* operon in *E. coli* Protein-Protein Interaction Network) and a human PPI with metadata on drug-protein associations provided for the Bio+MedVis Challenge 2023 (sec:deeprom).

5.3.1 Related Work

Simple node-link diagrams are the most commonly used visualization techniques for networks [192]. They are, e.g., popular for visualizing PPIs and are used by *STRING* [265] and the well-known network visualization tool *Cytoscape* [248]. Although node-link diagrams are conceptually intuitive and powerful for visual analysis, they quickly suffer from overdraw, layout problems, and clutter for larger networks [194].

Therefore, approaches beyond force-directed node-link diagrams, different layouts [262] including hierarchical layouts [12], on-node encodings [73], and hybrid network visualizations [9, 113] have been developed. For more in-depth insight into the vast literature on network visualization, refer to the recent state-of-the-art reports by Filipov *et al.* [84] and Nobre *et al.* [192] as well as the literature mentioned in Subsection 5.1.5.

To reduce the amount of clutter created by the edges, various approaches are used. Edges can be partially indicated [40] 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 [250]. Moreover, edge bundling can be used to use topological or semantic information to merge edges into bundles [119, 307]

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 [26, 250]. 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 [66]. In one such example, Vehlow *et al.* [279] 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 [113], chord diagrams [9] or customizable plots such as line- and bar-charts [73]. 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 [156]. 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.* [129]. 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 [219] circular glyphs [295], radial layouts [216, 224], linear layouts [88, 305] as well as combinations of the aforementioned approaches [249]. 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” [295]. 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 [65, 293, 300], especially in literature focused on ego-graph visualization [32, 79, 156]. Differently, EgoLines is an example of a technique applying non-circular layouts when comparing an ego-graph at different states of a dynamic network [305]. 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 [145]. The *STRING* database [174] 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 [265]. 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* [60] 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 [301], 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.

5.3.2 Approach

Based on the Bio+MedVis Challenge 2023 [145] and aided by the task taxonomy for graph visualization by Lee *et al.* [152], we identified the following tasks for the development of *ProtEGOnist*:

Overviews can provide a starting point for the analysis [252], especially in previously unexplored datasets 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 [105] 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.

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**).

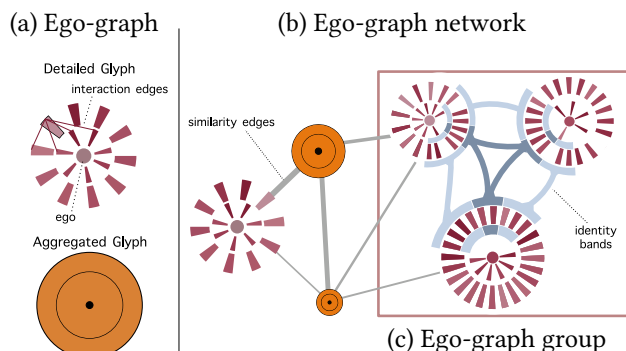


Figure 33: Ego-graphs and ego-graph networks, the concept of *ProtEGOnist*. (a) A single ego-graph can be visualized in detail or aggregated. The detailed view shows the ego in the middle of a circular graph layout. Degree-1 alters are placed on the inner circle, and degree-2 alters on the outer circle. 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.

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 33a). The detailed glyph (Figure 33a, 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 33a, 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 32). The font size is scaled with the size of the glyph, and the text labels are automatically discarded if the glyph is too small.

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 33b). 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 33c). 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.

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* [156], in which alters shared between two ego-graphs are connected using edges (Figure 34a). 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,

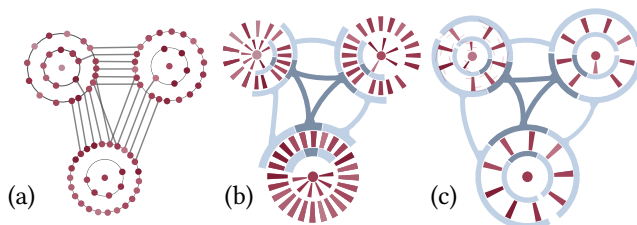


Figure 34: Different concepts for visualizing ego-graph groups: (a) First concept as submitted to the Bio+MedVis Challenge 2023. Nodes are encoded as circles and connected to the instances of the same node with identity edges. (b) Redesigned concept, where nodes are encoded as ring segments and identical instances are connected with identity bands. (c) Shared-only mode, where alters not shared with other ego-graphs in the group are filtered out.

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 34b), 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 34c, *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 Subsubsection Ego-graph Concept & Visualization Design. Furthermore, as individual identity edges are no longer drawn, more advanced sorting criteria could be introduced for the segments leading to a natural partitioning into subsections.

nodes: provided that the input 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 [142], 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 35). 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 32b, 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

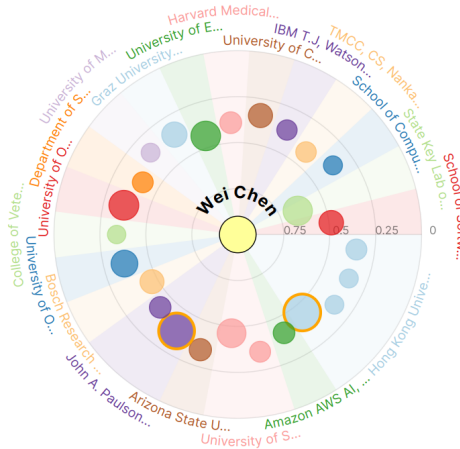


Figure 36: Similar ego-graphs to a reference ego-graph (radar center) in a co-authorship network classified by affiliation. The radar chart shows the 25 ego-graphs most similar to the ego-graph in the center. The distance to the center corresponds to the Jaccard index. In addition to the similarity, 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.

with the selected ego. This places the radar chart in close relation to the concept of *monadic exploration* [68]. The 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 [131]. 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

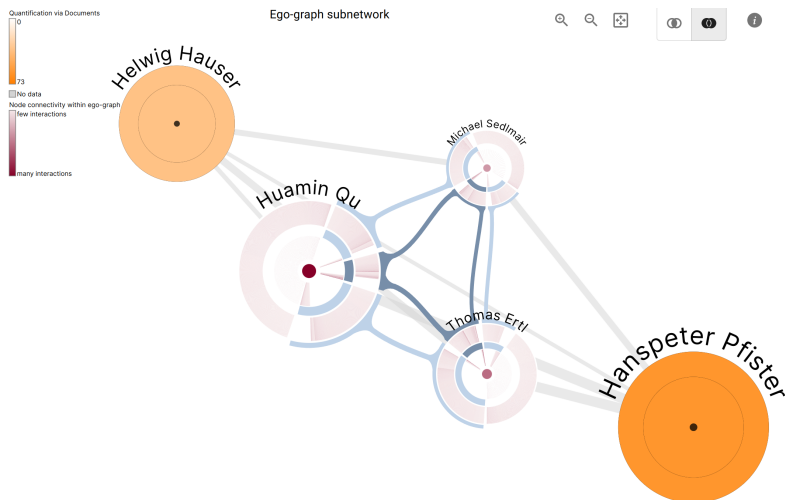
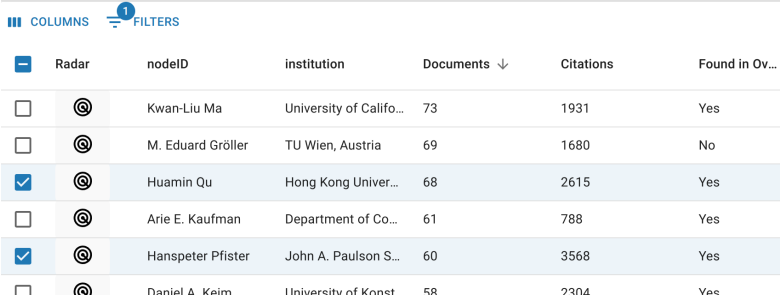


Figure 37: Ego-graph subnetwork of a co-author network with five ego-graphs. A group of three ego-graphs is shown in its detailed view. The width of the gray *similarity edges* encodes the similarity between ego-graphs outside ego-graph groups, while the blue *identity bands* link identical nodes within an ego-graph group.

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 36.

Ego-graph Subnetwork

Ego-graphs selected in the overview or the radar chart are visualized in the *ego-graph subnetwork* (Figure 32c), showing different levels of details of the respective ego-graphs. As mentioned in Subsubsection Ego-graph Concept & Visualization Design, the ego-graphs are initially visualized using the aggregated glyphs, but can be de-aggregated to the detailed glyphs on demand (Figure 37, **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 Subsubsection Ego-graph Concept & Visualization Design.



 COLUMNS	 FILTERS					
		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 38: Excerpt of the selection table showing the top 5 entries of the co-author dataset (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.

Selection Table

Groups of nodes in ego-graphs or intersections can be selected for investigation in the selection table (Figure 38), 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 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.

Implementation

We implemented *ProtEGOnist* as a web-based application with a client-server architecture. The server backend was written in Python using Flask [229]. The user interface and the visualizations in the frontend were mainly implemented in TypeScript using *React* [78], *Jotai* [135], *Material-UI* [182], and *D3* [29]. In the backend, we use the Python library *networkX* [104] for the extraction of relevant features from the graph structure. *ProtEGOnist* is available at “<https://protegonist-tuevis.cs.uni-tuebingen.de/>”.

Author	Avatar	Institution	Publications	Citations	InstitutionCountry	number_of_publications	number_of_edges	average_degree	degree_1_neighbors	degree_2_neighbors	degree_3_neighbors	degree_4_neighbors	total_publications	selected?	visited in this network?
☐		University of California	73	1931	USA	193	2284	4.0888912711	117	427	117	427	True	No	No
☐		Tel Aviv Academic	65	1880	Israel	174	2781	4.08321217884	143	439	143	439	True	No	No
☐		Harbin Engineering	46	2415	China	1746	6446	4.07949459767	246	1192	246	1192	True	No	No
☐		Harvard Medical School	41	186	USA	149	9178	4.07688494008	81	445	81	445	True	No	No
☐		Johns A. Packer S.	40	1368	USA	1218	4173	4.06817553521	143	1152	143	1152	True	No	No
☐		University of Konstanz	39	2384	Germany	193	3146	4.07096884616	124	474	124	474	True	No	No
☐		University of Bonn	34	1334	Germany	971	4012	4.067353447927	111	489	111	489	True	No	No
☐		University of Utah	47	1842	Germany	487	2178	4.04246942646	146	545	146	545	True	No	No
☐		State Key Lab of C.	47	1302	China	1467	4873	4.067695819176	128	547	128	547	True	No	No
☐		New York University	45	1688	USA	912	2427	4.116312871867	117	494	117	494	True	No	No
☐		Emmanuel University	42	1403	Emmanuel University	1247	1441	3.02777778244	91	239	91	239	True	No	No
☐		SO University of Bonn	42	1441	USA	441	1872	4.07077476747	129	274	141	274	True	No	No
☐		Department of C.	42	1323	China	1388	4624	3.026492192761	128	1179	128	1179	True	No	No
☐		University of Utah	41	617	University of Utah	1173	4676	4.0434748347483	122	462	122	462	True	No	No
☐		Department of C.	40	1257	USA	146	1235	3.162446767676	52	292	52	292	True	No	No

Figure 39: Table visualization for the co-authorship network. The rows have been sorted with respect to the number of published articles by each author.

5.3.3 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 dataset 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 datasets.

Co-author network

To showcase the usefulness of our *ProtEGOnist* approach for exploring social networks, we applied it to the Visualization Publications Dataset [121]. This dataset 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 [112] 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 35, **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 38, Figure 39), 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 40, **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 36). It also shows that the ego-graph of Chen contains researchers from institutions from all over the world.

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 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 [265] 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* [248]. Figure 41 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*

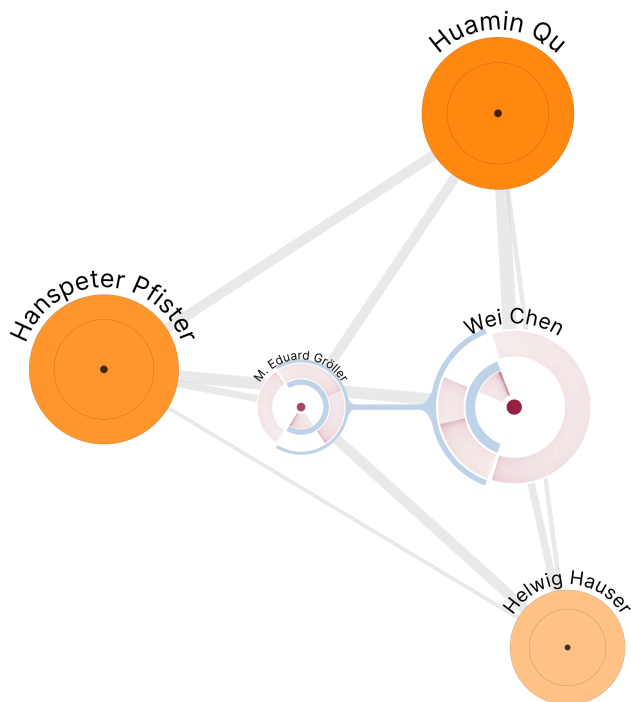


Figure 40: Ego-graph subnetwork visualization with a detailed ego-graph group of co-author networks of Gröller and Chen. The detailed glyphs reveal that Chen has not only a larger network in total but also a higher percentage of unique second-level co-authors.

StringApp [64]. 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 41). 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 41). Strikingly, we

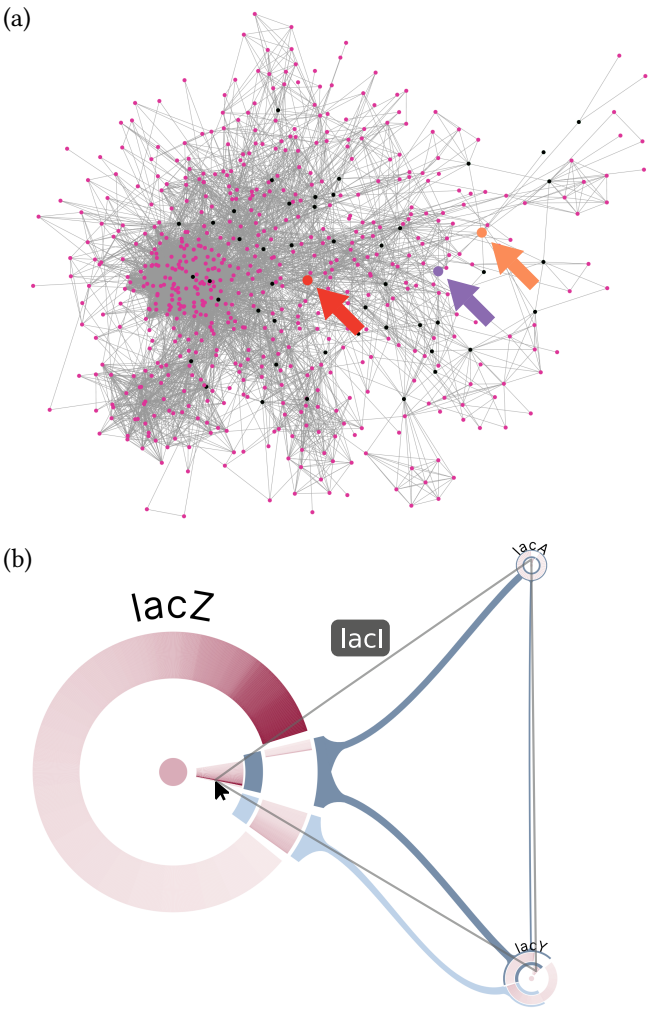


Figure 41: Visualizing the *lac* operon of *E. coli* using a node-link diagram created with Cytoscape (a). *lacZ*, *lacA*, and *lacY* are colored in ■, ■ and ■, respectively. Nodes of distance one are colored ■ while the ones of distance two are colored ■. The network consists of 653 nodes and 8,435 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.

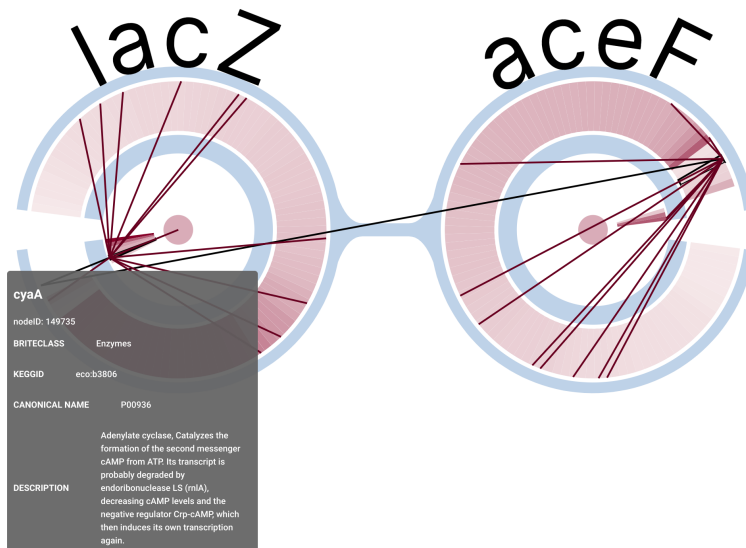


Figure 42: The ego-graph subnetwork of nodes shared by *lacZ* and *aceF*. When hovering over any degree-1 alters of the two egos, the identity edges are shown. This interaction is shown for *cyaA*. With this interaction, it can be seen that none of the degree-1 alters are found within the degree-1 alters of the other ego graph.

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 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 42).

Human DeeProM Protein-Protein Interaction Network

We used the current version of *ProtEGOnist* to analyze the dataset by Gonçalves *et al.* [98] 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.

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 43). 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* contained 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 44).

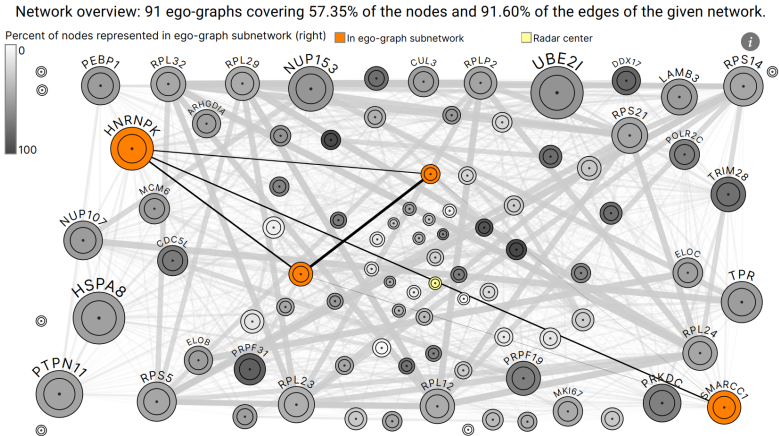


Figure 43: Network overview for the DeeProM dataset. The set of 91 ego-graphs is based on the proteins identified by Emanuel Gonçalves, Rebecca C. Poulos, Zhaoxiang Cai, Syd Barthorpe, Srikanth S. Manda, Natasha Lucas, Alexandra Beck, Daniel Bucio-Noble, Michael Dausmann, Caitlin Hall, Michael Hecker, Jennifer Koh, Howard Lightfoot, Sadia Mahboob, Iman Mali, James Morris, Laura Richardson, Akila J. Seneviratne, Rebecca Shepherd, Erin Sykes, Frances Thomas, Sara Valentini, Steven G. Williams, Yangxiu Wu, Dylan Xavier, Karen L. MacKenzie, Peter G. Hains, Brett Tully, Phillip J. Robinson, Qing Zhong, Mathew J. Garnett, and Roger R. Reddel. [98] as part of the 108 most-prominent protein-drug associations of their dataset. 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.

The lesser connected protein *SMARCC1* has been identified as a suppressor in some types of cancers [297], while the others act as possible drug targets or cancer biomarkers [288, 299, 303]. All three proteins are associated with the BRITE class *Spliceosome*, i.e., these proteins are involved in the maturation of mRNA before translation [284]. This association is even more prominent when inspecting the proteins *PPIH* and *RBM39* using the radar chart (Figure 45, R4).

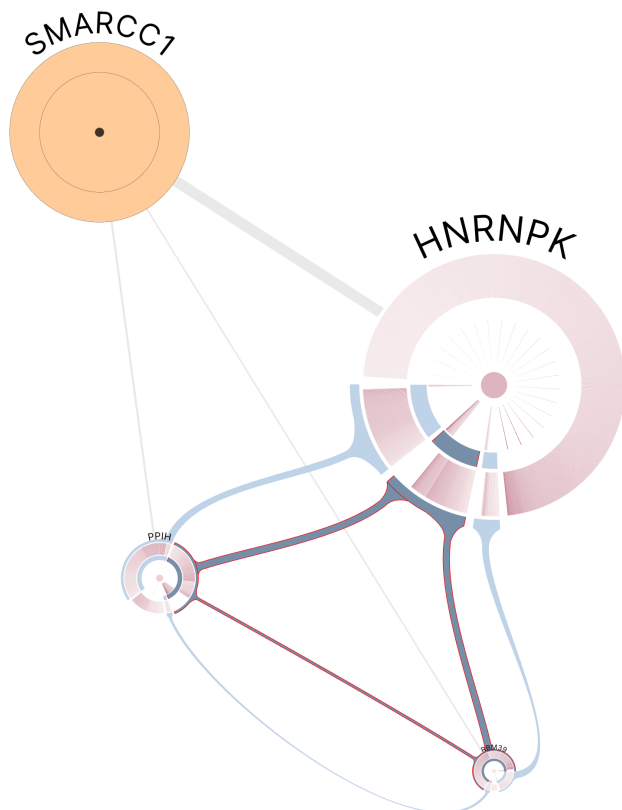


Figure 44: Ego-graph subnetwork for all proteins of the overview network associated with the drug ARA-G. The three most highly connected ego-graphs (*PPIH*, *HNRNPK*, and *RBM39*) are shown as an ego-graph group. The intersection between all three ego-graphs has been selected (red outline).

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.

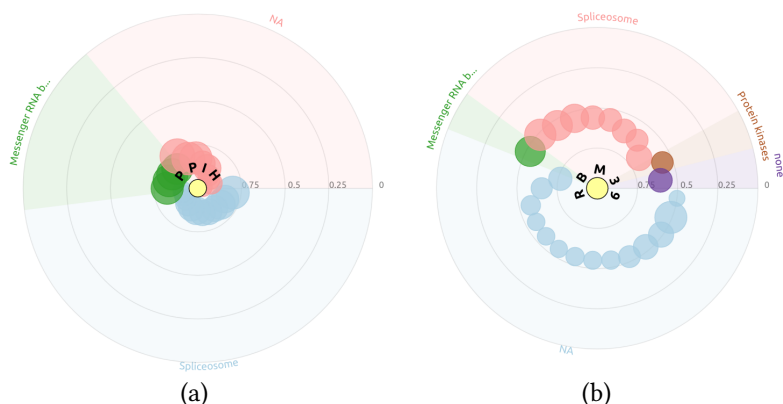


Figure 45: Radar charts for the proteins *PPIH* (a) and *RBM39* (b). The closer the circles are to the center of the radar chart, the more similar the ego-graphs.

Expert Feedback

Due to the positive outcome of the Bio+MedVis Challenge 2023 [145], we contacted the authors of the DeeProM dataset 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 dataset using *ProtEGOnist*. Subsequently, we provided a structured questionnaire of ten questions based on the System Usability Scale (SUS) framework [39] and further open-ended questions.

The expert that evaluated our tool had explored similar datasets using visualizations provided by *STRING* [265] and *Reactome* [77]. *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.

5.3.4 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 datasets 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 chart 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 [43], computer networking [296], or transportation networks [150].

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* [305], among others [88, 249], 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* [156] 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.

5.3.5 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 [174]) 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 [60]. 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 41) 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.

5.4 Summary and Conclusion

Both approaches, *VisMOP* and *ProtEGOnist*, are EDA applications to explore biological networks that are too large for static visualization. While they share this aim, they do have notable difference regarding the underlying data and how the network is visualized, especially how they differ both in type of network and what their nodes and links represent. This is mainly due to their different intended application. *VisMOP* is intended to allow users to explore the network and find similarly regulated metabolic-pathways by applying clustering to metadata derived from multi-omics measurements and investigating these pathways in the context of the complete pathway network. In contrast, *ProtEGOnist* allows to explore any kind of small-world network, such as the *STRING* network [265], focussed on nodes-of-interest and the similarity of their immediate neighborhoods.

With different application domains, of course, come different data sources and data structures used for the network at the core of the application. The network data in the case of *VisMOP*, *Reactome* [93], is an ensemble of trees, i.e., multiple hierarchical components, with some links between the nodes of the components. *ProtEGOnist* on the other hand needs a network fulfilling the *small-world property* to be used meaningfully. In its inception *ProtEGOnist* was build for the visualization of PPIs, using the *STRING*-databse [265] as source where links represent association based on information from metabolic-pathway databases, like *Reactome* [93] and *KEGG* [130] and other metrics like genomic neighborhood. This makes the links in *ProtEGOnist* a lot less easy to understand as they aggregate many kinds of information, while links in *VisMOP* are always hierarchical or represent real links, i.e. physical continuations of a pathway. However, this also means that links in *ProtEGOnist* contain a lot more information, which can be leveraged for exploration.

This difference in what the links are in the original data and the underlying analysis goal influence the role of the links in the exploration of the network. As the primary analysis goal of *VisMOP* is to find similarly regulated pathways, and the information encoded by the links is not par-

ticularly relevant for this task, a less prominent role is chosen for them. As many types of networks layouts are defined by the links between the nodes, this also influences the layout choice. Therefore, *VisMOP* uses a circular layout on the outside for the alphabetically ordered roots of the pathway components and a cluster based positioning that is exclusively performed on the metadata. Links are only shown on demand for a detailed analysis. For *ProtEGOnist*, however, the topological information is the core aspect that is visualized. Thus, it uses a force-directed, hybrid, layout that is exclusively derived from topological data.

While in both approaches the nodes are aggregates of information, both the information aggregated and the manner it is displayed, differ between *VisMOP* and *ProtEGOnist*. In *VisMOP* the nodes are glyphs representing a metabolic-pathway and its multi-omics changes. In *ProtEGOnist*, on the other hand, nodes aggregate an ego-graph around a protein of interest in a *meta-node*. This aggregation can be undone interactively in the main view of *ProtEGOnist* to reveal a subnetwork showing details about the subnetwork and other meta-nodes. *VisMOP* also offers inspection into the aggregation, but does so using a linked, superimposed view showing the metabolic network in the layout generated by *Reactome* [93]. Thus, both methods offer interactive methods to undo this aggregation to explore the complete underling network in its original detail in a focussed manner.

One drawback of being domain agnostic is, that in contrast to *VisMOP* where the domain-specific information, namely multi-omics-regulation values, are encoded as glyphs on the nodes, *ProtEGOnist* only uses general topological information for a similar purpose. While it would be possible to use domain-specific values in *ProtEGOnist*, it would mean to either specialize the approach for a specific domain or make it possible for the users to configure the approach. This could be either in a library or interactively. On the flip side, *VisMOP* is highly specialized, which makes its application area very narrow. Here, a generalization could be devised to be applied to any kind of network that visualizes a hierarchy of aggregations. In this case, challenges would be the use-case specific

detail visualizations as well as the generation of the values used for clustering of the aggregations.

Together, these approaches cover a wide variety of aspects, both on what kind of data they visualize and also how they do visualize this information. Importantly, they illustrate different aspects on how network visualization can be leveraged for EDA. Among these things are the choice of a level-of-detail, i.e., the level of aggregation for data in the nodes and links. The type of data influences both the network layout and if the focus of the visualization should be on nodes, on links or both. Certainly, there are other aspects that are not covered, but that are crucial to unlocking the full potential of network-visualizations for EDA. Here, the recent *State of the Art Report* (STAR) [70], gives more insight into network-visualizations for biological data and shows how other approaches integrate them in an EDA-setting.

The background features a series of concentric circles and radial lines in a light gray color. The lines intersect at various points, creating a network-like pattern. A large, solid gray circle is positioned in the lower right quadrant, serving as a focal point. The overall design is minimalist and modern.

6

DISCUSSION

This dissertation shows the development of multiple EDA applications, which can be used to gather hypotheses in a personalized medicine scenario like the one outlined in Section 2.1. To tackle the first part of that scenario, when a patient is admitted to the ICU, presenting potentially life-threatening symptoms, we developed *ICUVA*. *ICUVA* is an exploratory and holistic approach to patient vital surveillance. It allows medical personnel to easier pinpoint substantial changes in a patient's homeostatic system. Furthermore, *ICUVA* can then be used to find the vital parameters causal for that change allowing a more educated approach when characterizing the primary pathology. Concurrently, we contributed to a scoping review, evaluating the potential, and the existing gaps, of visualization when comparing EHR data of single or multiple patients with a multitude of other such data sets. Approaches tackling these questions could bolster hypotheses regarding the pathology of a patient even further.

Answering the second challenge of the abovementioned scenario, finding biochemical causes for the pathology, we developed a method to help laboratory scientists in finding molecular changes and patterns in multi-omics data. Such changes are especially interesting in highly variable diseases, like cancers, or rare diseases where the route of treatment heavily depends on the genetic or molecular signatures. Here we developed *VisMOP*, an approach to connect experimental multi-omics measurements with metabolic pathways. With this approach, researchers can find similarly regulated pathways on a trans-omic level, which in turn can shed light on underlying changes in physiological processes. Additionally, this can also yield serendipitous, novel insights, like new molecules—at least in the context of the research problem—warranting further research.

At its core, *VisMOP* utilizes and visualizes a graph structure, a metabolic network. As such, this knowledge about networks was applied and further deepened while working on a state-of-the-art report [70] that navigates the large research space of graph visualizations in biomedical data, including metabolic networks and many more.

This deepened knowledge helped us in the development of *ProtEGOnist*, in which relational data, primarily of protein-protein-interactions, was visualized. With *ProtEGOnist*, researchers can use proteins of interest as an entry point to find related proteins within a PPI-network. In our scenario, an entry point could be a protein identified using *VisMOP*. Finding related proteins in such a manner can open up avenues for targeted treatment, as not only the original protein of interest can be targeted, but also related ones that act at other points in the metabolic cascade.

Yet, identifying potential targets is just the first step in finding a treatment. Oftentimes there is no known drug specifically targeting a protein. The second step can be found in the discipline of computer-based drug-design. Here, virtual protein-ligand-docking is performed, that is the computational screening of small molecules concerning their capacity to bind to a specific protein. This then yields a set of potential drugs that can be screened in further (*in vivo*) studies. I contributed to InVADo [244], an EDA-suite specifically aiming at visualizing the results of such docking experiments.

Concluding, this work shows how EDA applications, developed in the context of a variety of personalized-medicine scenarios can yield starting points for future research analysis. These applications, can be domain-specific tools as shown with *ICUA* or *VisMOP* but also generalized approaches. As an example, we generalized *ProtEGOnist* to be applied to all types of small world networks in biomedical data but also in other field, making it domain agnostic. With this it could be applied to a field we did not explicitly designed it for, like epidemiology. Here, researchers could visualize contact networks to track and elucidate the spread of an infectious disease. In general, these examples showcase the power EDA and especially visualization in scenarios in which simple answers are hard to come by.

6.1 Lessons Learned

Besides developing the concepts and prototypes for EDA of biomedical data, this thesis produces several insights and lessons learned for visualization design. The first insight concerns the nature of the data itself.

6.1.1 Considering uncertainty is difficult

As indicated in the background, the data used in biomedical research is generated using numerous techniques, each introducing different levels of uncertainty. In the medical domain much of the data are EHRs, which often feature missing data, while the data in the biological domain is mostly gathered in laboratory experiments. Some of the introduced uncertainty stems from laboratory work, which is imperfect by nature owing to the high variance in the subjects. Additionally, the techniques used also are often not perfect. For example, when employing mass-spectrometry it is easy to measure incorrect amounts of molecules or miss them altogether [90].

Furthermore, many types of research are relying on other resources, like reference databases. These databases are reliant on experiments themselves and thus are imperfect by extension. This uncertainty needs to be considered at many points during analysis. For one when just interpreting the result of a single step, like a significance test, but just as important it is to consider it when designing an EDA approach.

Considering uncertainty adds another dimension to data that would require even more visualization design and engineering work. This compounds with another major insight that concerns the very nature of the development of EDA visualizations, or complex visualizations in general. That is the fact that domain-specific EDA visualizations require a lot of functionality in addition to the core functionalities of the approach and thus extensive amounts of work, knowledge and resources.

6.1.2 EDA applications are complex, so is their development.

The primary aim of this thesis is to show the usefulness of EDA applications in the biomedical domain. The successfulness, is shown in the

systems described in the previous chapters. EDA applications are not *free*, however, and developing them is in fact very complex and time-consuming. Some specific aspects of these complexities are already described in the respective sections. In this section more general aspects when designing EDA applications are discussed.

One aspect is that intimate knowledge of the domain is needed to analyze the needs of potential analysts. In the case of biomedical data, the domains are particularly large as well. A practicing physician has different data and a different need for analysis as a medical researcher or a biologist researching, for example, molecular mechanisms of cancer development. Thus, designers and developers of the visualization either need extensive previous knowledge on the subject or must communicate and cooperate extensively between domain experts and said personnel. In any case, this step should be thorough to allow for the generation of specific questions to the data. Because, as described in Chapter 3, visualization design relies heavily on systematically specifying the tasks and functionalities that should be supported. Thus, if, previous to implementation and design, the goal and the aspect of the data that needs visualization are only poorly described, it becomes a challenge to generate a visualization that yields helpful and novel insights into the data.

A second aspect is the need of the designers to understand how the data that should be visualized came to be. While a visualization designer could conceivably consider the data a black-box and just take it the way the data presents itself, this has drawbacks. Primarily this causes the designer not to consider possible imprecision of the data and how to judge them. When manually recording vital data, for example it might easily occur, that circumstances prohibit the scheduled recording of a specific data point. While this is decreasing the precision of the dataset, this is an unavoidable facet of the domain and does not put into question the validity of the whole dataset. On the other hand, if a long-term blood pressure measurement system fails to record any data for a given amount of time, to then return to recording it might be prudent to question the validity or precision of the system. This means, that a specific undesired

behavior might be annoying but still perfectly normal in one type of data in a given scenario, it might be an important red flag in another scenario.

Furthermore, data visualization as a whole tends to underestimate the uncertainty of the data, which often results in a visual language that assigns relative or absolute truth to a given visualization. In PPI-Networks, edges often make people assume that there is a definite interaction between the two proteins, as the edge between two nodes is binary—there or not there. Nevertheless, any researches using these would need to know what kind of evidence has been used to assign those edges. Having this intimate knowledge of how the data is generated, however, can help the designer to consider this uncertainty in visualization design.

A third aspect for complex EDA systems, is often the linkage of the data with additional background information from databases and the like. In bioinformatics, this is often the colloquial *ID matching* phase. Here, the data that is measured in experiments is often annotated with a specific system, sometimes custom to a lab, tying it back to the knowledge about the specific field. A classic example here is a gene ID. Depending on several things, among others, the researched organism, the used ID system can vary. If designers then want to enrich the data with other information gathered from other databases these IDs need to be matched to result in the correct hit, and with it the matching information. This causes a designer to engineer a visualization system that is specifically tailored to one specific pipeline, which oftentimes is then also only used at one given domain lab. It has to be noted, however, that there are considerable efforts to standardize the ID systems, for example by providing adapter tools and also to establish standard pipelines in frameworks like *Nextflow* [266].

The other way would be to design a flexible data loading system that considers different types of input data, like different ID systems, to be linked with other variable additional (meta-)data. This solution is obviously quite a lot more labor-intensive. Besides *ID matching* other data-wrangling tasks must be performed here as well, and these are oftentimes both labor-intensive and also require extensive knowledge and abilities in specific domains of data science. Of course the designer of

the visualization might also just limit the application to a specific data-type, which reduces development effort, but that puts the burden on the researcher, which then might limit the appeal of the approach.

A fourth and maybe most vital aspect, the actual visualization has to be designed. And, as the center of visualization research, this includes the steps generally performed when designing any type of visualization. However, for most EDA applications, and especially dashboard-like applications, designers need to develop more than one singular visualization type. They often rely on a set of linked visualizations to fulfill several steps of the EDA, for example, having distinct visualizations for overviews and details on demand.

Designing a visualization is just one part of the story. While a description of a visualization system might sometimes be sufficient, visualization designers are almost always required to not only design but also at least develop a prototypical application of the system. With this aspect in the development of an EDA approach opens the floodgates on a whole load of accompanying duties and problems that need to be sorted out. With any software project, there are myriad platforms and package choices that need to be taken. These choices must be informed by data about target users, about how computationally intensive data-wrangling tasks are, and if any specific pre-existing tools or packages need to be used to implement the intended designs.

Ultimately, any scientific visualization approach should be evaluated as well to have an idea of how useful a given tool is for a given task, in this case for EDA. Here, in the sixth step, multiple approaches can be taken. The first one is a full-scale user study with a large amount of participants allowing for the quantitative analysis and comparison with different approaches. For this, however, all other aspects of the tool need to be stable and tested as not to taint possible results. Furthermore, such studies mean considerable effort comparable to the implementation and the write-up of the original publication. The other approach is to perform small-scale user feedback sessions. Here a small set of domain experts can either explore a tool by themselves or can be shown the functionality

of a tool in a pair session with one or more of the developers. The actual execution of such a session can be very variable, however.

When dealing with these problems, it is also possible to abstract more heavily. Usually, this then results in solutions for a given sub-problem that can be applied to other research problems. As an example, a data-wrangling method developed for a specific EDA application can be, given if the solution is implemented in an abstract fashion, applied and published for other problems like statistical algorithms.

While this sounds very appealing in theory, the real-life application of this can be rather complicated. A primary problem is quite apparent, that, in an academic setting resources are often limited, making academia not particularly attractive for software engineers, given the competitive salaries in industry. So it is natural to ask what could be done in an academic setting to allow for more systematic development of complex tools, as this applies to all larger user-oriented software beyond just EDA, when just throwing more money at it is not a viable option. And here finding synergistic cooperation between academic working groups can be used as a substitute for additional personnel. Ideally, people with knowledge in the domains associated with the abovementioned sub-problems can work together to solve the problem in a more modular manner. Of course, this comes with other drawbacks, like the additional overhead introduced by the increased amount of stakeholders. More importantly, PhD programs oftentimes require multiple publications as a primary author and as such a dedicated research question. So in order for each PhD student to fulfill this requirement, it would require quite a lot of concurrent projects. As such it would be naïve to assume that each stakeholder can fully focus on their task, making progress a lot slower. In reality, not every PhD student has their *own* cooperative research project. The existing ones are then often sidelined, as PhD students focus on their solo-projects required for them to advance. Furthermore, it is very difficult—or particularly time-consuming and unlikely—to find experts in each subdomain willing to cooperate on a given problem.

Ultimately it is also possible to select only a single aspect to research using the steps mentioned above. In effect, instead of developing a

complete application, a single researcher could just focus on developing, for example, a data-wrangling algorithm for a specific domain problem or a visualization design tackling a specific task. Here PhDs can focus on a single task while providing the minimum extraneous work to give a proof-of-concept. A benefit of this would be that, when finished, such developments can be combined with various other components to generate pipelines and tools, such as EDA visualizations.

In an academic setting, this can be problematic as, depending on the domain problem, the sub-problems can become quite granular. They can, in fact become so granular that they lose all connection to the application domain, making it difficult to publish their solutions in domain journals. For example when visualizing metabolic networks, sub-problems can involve finding efficient layout algorithms or optimizing navigation in a large network. These problems are then no longer specific to the original domain which spawned the research. For biomedical visualization-research this means that some respective publication venues would not publish solutions to those sub-problems at all.

The previous paragraphs outline theoretical ways to make the complex work of designing and implementing EDA applications in an academic setting more efficient. While the described ideas and observations stem from personal experience during the creation of this work and its projects they echo observations by working groups aiming to improve the quality of visualization software developed in academia [223].

6.1.3 Compromises We Made

There are some compromises that can be made to contribute to the field in an academic setting. Examples of these compromises are found in the work that constitutes this thesis. As mentioned earlier, it is possible to just limit the scope of development, to still make meaningful contributions to the scientific field of EDA. In fact in two of the tools, *ICUVA* and *VisMOP*, we mention that they are developed primarily as a prototypical application. In other words, they should primarily act as a proof of concept for a visualization approach. Ultimately, however, any EDA-approach is meant to be applied to data to develop hypotheses and gain

insights into data. By exposing this idea to the academic- and general-public, those approaches can be developed into full-fledged tools. So here, what is missing with the presented approaches to be used as a tool and not only to serve as a prototype will be briefly discussed.

The three presented approaches differ regarding how much changes or additional features would be required for it to be used as an actual tool. *ProtEGOnist* is the most complete *tool*. Because *ProtEGOnist* several core hallmarks are already present. The most basic requirement for a tool is accessibility, i.e. that users can either access the source-code or a running version of the tool. *ProtEGOnist* is both accessible online, and it is easily installed locally as well, owing to a complete docker bundle.

In many cases, to be more than a demo-application users need to be able to upload their own data. This is a self-explanatory aspect as, bar very specific use cases, user defined data-sets are the object of investigation during EDA. *ICUVA* and *VisMOP* require quite specific data input as they have been developed as a prototype with limited amount of datasets. *ProtEGOnist* however supports multiple types of graph representations, like *GraphML* or adjacency-lists. To enrich the visualization, *ProtEGOnist* also supports arbitrary meta-data that is associated with the nodes. Furthermore, *ProtEGOnist* is domain agnostic, which means, that it does not have any specific further requirements that must be fulfilled by either the meta-data or the graph data itself.

Many EDA-scenarios require quite a lot of software development for interaction and presentation alone. For these functionalities, we utilized a wide array of programming libraries offering ready-made functionalities. This echoes a pivotal tenet in programming: “Reuse instead of reimplement.” Ideally, for each part of the software implementation would exactly realize the design goals. However, oftentimes these libraries, while configurable, do not allow for the exact realization of all goals. We thus, choose to allow deviation in non-core aspects. This primarily includes UI, interaction, and rendering libraries, which are, of course, still important building blocks of an EDA-approach. For most EDA-interactions this is sufficient, however, some specific core-application areas needed considerable effort to support workflows that are not

readily supported by mainstream libraries. This includes for-example handling of non-standard data-types, e.g. Reactome [93] pathways used in *VisMOP*, or the ego-layout in *ProtEGOnist* among others.

Neither *ProtEGOnist* nor any of the other presented approaches fields a full-fledged user study. One pivotal problem when designing applications for domain specialists is that quantitative user studies require a large amount of participants. Nevertheless, recruiting so many domain scientists for a study is difficult. In general, novel visualizations are compared to a well-known alternative. For EDA applications this is difficult, as a comparison would require both tools to answer the identical research question.

Because of this, we performed a smaller-scale user-feedback session for *ICUVA*, with a few participants and guided exploration of the prototype, an approach also used in *InVADo* in a more formalized setting. With *VisMOP* we decided for a qualitative evaluation of the prototype by the collaborating domain scientists, both during development as a feedback-loop as well as a final evaluation before publication. A similar approach was taken with *ProtEGOnist*. As *ProtEGOnist* is domain agnostic there is the most potential for quantitative evaluation, which can be explored in the future.

For *ProtEGOnist* we made on core compromise to streamline the development process. This is the limitation of the scope of *ProtEGOnist* just on exploration of a network based on nodes of interest, without any domain-specific requirements. This allowed to focus on the visualization aspect, which we deemed most central for the project. Additionally, *ProtEGOnist* was developed in a two-step process with a development team of three developers. In the first step, which was based on the *BioMedVis-Challenge 2023*, the basic concept was designed. The second step was the development into a functional tool as part of the publication at the *EuroVis 2024*. This made it a lot easier to follow the divide-and-conquer approach described above.

These *lessons learned* are not unique to these specific projects but can be found all over academic visualization reasearch. It thus seems imperative that we, as a research field, try to take these lessons, especially in their

published form, to heart at alleviate the problems as good as possible instead of letting each generation of researchers learning them again and again.

6.2 Conclusion

While EDA is applied in almost every type of research, an area in which it particularly shines in is biomedical research. Biomedical domain scientist generate a lot of data in experiments designed with a specific theory and hypothesis in mind. After this question is tackled, many scientists realize that the data might still contain valuable and novel information. This is of course due to the fact, that most experiments in biological systems cannot be narrowed in a way to only generate the data, which is needed to describe the phenomenon in question. Experiments, especially today, produce a lot of data while being oftentimes expensive, both in labor and cost. Thus, EDA is a tool to responsibly exploit the time and monetary resources in biomedical sciences.

Additionally, this dissertation also outlines how EDA can be used in scenarios where data is recorded not as part of a different experiment, but where data is recorded in the hopes that EDA can give an entry point for research at all. The presented tools can be used for both these scenarios, i.e., exploiting already evaluated data further and recording data for the explicit purpose of exploration. Yet, unlike the scenario outlined in Section 2.1, which makes use of both approaches, much of the data on which the tools were developed was originally recorded for other purposes.

And while it is enticing to throw EDA at every data available—which in theory would be the responsible thing to do—in many cases such EDA-approaches are single-purpose and tools specific to research (sub-)domains. But these custom-build tools require considerable development effort to only serve a very specific analysis, which might only be performed a single digit number amount of times. In such cases, it is often not be feasible to develop an EDA application. However, when an EDA-approach can be generalized to fit a wide array of different domains

and problems it can be very impactful and might help push the research space more than a single tailor made algorithm would.

Bibliography

- [1] Vue.js - The Progressive JavaScript Framework | Vue.js. Retrieved May 3, 2026 from <https://vuejs.org/>
- [2] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. TensorFlow: A System for Large-Scale Machine Learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 2016. USENIX Association, Savannah, GA, 265–283. Retrieved from <https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi>
- [3] Moataz Abdelaal, Nathan D. Schiele, Katrin Angerbauer, Kuno Kurzhals, Michael Sedlmair, and Daniel Weiskopf. 2022. Comparative Evaluation of Bipartite, Node-Link, and Matrix-Based Network Representations. *IEEE Transactions on Visualization and Computer Graphics* (2022), 1–11. <https://doi.org/10.1109/TVCG.2022.3209427>
- [4] W. Aigner, S. Miksch, H. Schumann, and C. Tominski. 2011. *Visualization of Time-Oriented Data*. Springer, London. <https://doi.org/10.1007/978-0-85729-079-3>
- [5] M. Ali, M. Jones, X. Xie, and M. Williams. 2019. TimeCluster: Dimension Reduction Applied to Temporal Data for Visual Analytics. 35, 6 (2019), 1013–1026. <https://doi.org/10.1007/s00371-019-01673-y>
- [6] Waqar Ali, Tiago Rito, Gesine Reinert, Fengzhu Sun, and Charlotte M. Deane. 2014. Alignment-Free Protein Interaction Network Comparison. *Bioinformatics* 30, 17 (September 2014), i430–i437. <https://doi.org/10.1093/bioinformatics/btu447>

- [7] J C Alwine, D J Kemp, and G R Stark. 1977. Method for Detection of Specific RNAs in Agarose Gels by Transfer to Diazobenzyloxymethyl-Paper and Hybridization with DNA Probes. *Proceedings of the National Academy of Sciences* 74, 12 (December 1977), 5350–5354. <https://doi.org/10.1073/pnas.74.12.5350>
- [8] Gennady L. Andrienko and Natalia V. Andrienko. 1999. Interactive Maps for Visual Data Exploration. *International Journal of Geographical Information Science* 13, 4 (June 1999), 355–374. <https://doi.org/10.1080/136588199241247>
- [9] Lorenzo Angori, Walter Didimo, Fabrizio Montecchiani, Daniele Pagliuca, and Alessandra Tappini. 2019. ChordLink: A New Hybrid Visualization Model. *Graph Drawing and Network Visualization 11904*, 276–290. https://doi.org/10.1007/978-3-030-35802-0_22
- [10] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. 1999. OPTICS: Ordering Points to Identify the Clustering Structure. *ACM SIGMOD Record* 28, 2 (June 1999), 49–60. <https://doi.org/10.1145/304181.304187>
- [11] F. J. Anscombe. 1973. Graphs in Statistical Analysis. *The American Statistician* 27, 1 (February 1973), 17–21. <https://doi.org/10.1080/00031305.1973.10478966>
- [12] Daniel Archambault, Tamara Munzner, and David Auber. 2007. TopoLayout: Multilevel Graph Layout by Topological Features. *IEEE Transactions on Visualization and Computer Graphics* 13, 2 (March 2007), 305–317. <https://doi.org/10.1109/TVCG.2007.46>
- [13] Ira Assent. 2012. Clustering High Dimensional Data. *WIREs Data Mining and Knowledge Discovery* 2, 4 (July 2012), 340–350. <https://doi.org/10.1002/widm.1062>
- [14] B. Bach, P. Dragicevic, D. Archambault, C. Hurter, and S. Carpendale. 2017. A Descriptive Framework for Temporal Data Visualizations Based on Generalized Space-Time Cubes. 36, 6 (2017), 36–61. <https://doi.org/10.1111/cgf.12804>
- [15] Benjamin Bach, Conglei Shi, Nicolas Heulot, Tara Madhyastha, Tom Grabowski, and Pierre Dragicevic. 2016. Time Curves: Fold-

- ing Time to Visualize Patterns of Temporal Evolution in Data. *IEEE Trans. Vis. Comput. Graphics* 22, 1 (2016), 559–568. <https://doi.org/10.1109/TVCG.2015.2467851>
- [16] David Baltimore. 1970. Viral RNA-dependent DNA Polymerase: RNA-dependent DNA Polymerase in Virions of RNA Tumour Viruses. *Nature* 226, 5252 (June 1970), 1209–1211. <https://doi.org/10.1038/2261209a0>
- [17] M. Balzer and O. Deussen. 2005. Voronoi Treemaps. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, 2005. IEEE, Minneapolis, MN, USA, 49–56. <https://doi.org/10.1109/INFVIS.2005.1532128>
- [18] Albert-László Barabási and Zoltán N. Oltvai. 2004. Network Biology: Understanding the Cell's Functional Organization. *Nature Reviews Genetics* 5, 2 (February 2004), 101–113. <https://doi.org/10.1038/nrg1272>
- [19] Fabian Beck, Michael Burch, Stephan Diehl, and Daniel Weiskopf. 2017. A Taxonomy and Survey of Dynamic Graph Visualization. *Computer Graphics Forum* 36, 1 (2017), 133–159. <https://doi.org/10.1111/cgf.12791>
- [20] Michael Becker-André and Klaus Hahlbrock. 1989. Absolute mRNA Quantification Using the Polymerase Chain Reaction (PCR). A Novel Approach by a PCR Aided Transcript Titration Assay (PATTY). *Nucleic Acids Research* 17, 22 (1989), 9437–9446. <https://doi.org/10.1093/nar/17.22.9437>
- [21] Mouna Bencheikroun, Baptiste Chevallier, Vincent Zalc, Dan Istrate, Dominique Lenne, and Nicolas Vera. 2023. The Impact of Missing Data on Heart Rate Variability Features: A Comparative Study of Interpolation Methods for Ambulatory Health Monitoring. *IRBM* 44, 4 (August 2023), 100776. <https://doi.org/10.1016/j.irbm.2023.100776>
- [22] Brian J. Bender, Stefan Gahbauer, Andreas Luttens, Jiankun Lyu, Chase M. Webb, Reed M. Stein, Elissa A. Fink, Trent E. Balius, Jens Carlsson, John J. Irwin, and Brian K. Shoichet. 2021. A Practical

- Guide to Large-Scale Docking. *Nature Protocols* 16, 10 (October 2021), 4799–4832. <https://doi.org/10.1038/s41596-021-00597-z>
- [23] Derrick A. Bennett. 2001. How Can I Deal with Missing Data in My Study?. *Australian and New Zealand Journal of Public Health* 25, 5 (October 2001), 464–469. <https://doi.org/10.1111/j.1467-842X.2001.tb00294.x>
- [24] Jürgen Bernard, Eduard Dobermann, Anna Vögele, Björn Krüger, Jörn Kohlhammer, and Dieter Fellner. 2017. Visual-Interactive Semi-Supervised Labeling of Human Motion Capture Data. *Electronic Imaging* 29, 1 (January 2017), 34–45. <https://doi.org/10.2352/ISSN.2470-1173.2017.1.VDA-387>
- [25] Jürgen Bernard, Nils Wilhelm, Maximilian Scherer, Thorsten May, and Tobias Schreck. 2012. TimeSeriesPaths: Projection-based Explorative Analysis of Multivariate Time Series Data. In *Journal of WSCG20*, 2012. 97–106.
- [26] Mustafa Bilgic, Louis Licamele, Lise Getoor, and Ben Shneiderman. 2006. D-Dupe: An Interactive Tool for Entity Resolution in Social Networks. In *2006 IEEE Symposium On Visual Analytics Science And Technology*, October 2006. IEEE, Baltimore, MD, USA, 43–50. <https://doi.org/10.1109/VAST.2006.261429>
- [27] Brett J. Bordini. 2022. Undiagnosed and Rare Diseases in Critical Care. *Critical Care Clinics* 38, 2 (April 2022), 159–171. <https://doi.org/10.1016/j.ccc.2021.12.002>
- [28] I. Borg and P. J. F. Groenen. 2005. *Modern Multidimensional Scaling: Theory and Applications* (2nd ed.). Springer, New York. <https://doi.org/10.1007/0-387-28981-X>
- [29] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. 2011. D³ Data-Driven Documents. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (December 2011), 2301–2309. <https://doi.org/10.1109/TVCG.2011.185>
- [30] Jessica C. Bowman and Loren Dean Williams. 2023. Nucleic Acids. *Encyclopedia of Astrobiology*, 2096–2103. https://doi.org/10.1007/978-3-662-65093-6_1079

- [31] Marion M. Bradford. 1976. A Rapid and Sensitive Method for the Quantitation of Microgram Quantities of Protein Utilizing the Principle of Protein-Dye Binding. *Analytical Biochemistry* 72, 1–2 (May 1976), 248–254. [https://doi.org/10.1016/0003-2697\(76\)90527-3](https://doi.org/10.1016/0003-2697(76)90527-3)
- [32] Ulrik Brandes and Christian Pich. 2010. More Flexible Radial Layout. *Graph Drawing* 5849, 107–118. https://doi.org/10.1007/978-3-642-11805-0_12
- [33] Matthew Brehmer and Tamara Munzner. 2013. A Multi-Level Typology of Abstract Visualization Tasks. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (December 2013), 2376–2385. <https://doi.org/10/f5h3q4>
- [34] N. Brich, T.A. Harbig, M. Witte Paz, K. Nieselt, and M. Krone. 2024. ProtEGOnist: Visual Analysis of Interactions in Small World Networks Using Ego-graphs. *Computer Graphics Forum* 43, 3 (June 2024), e15078. <https://doi.org/10.1111/cgf.15078>
- [35] N. Brich, N. Schacherer, M. Hoene, C. Weigert, R. Lehmann, and M. Krone. 2023. visMOP – A Visual Analytics Approach for Multi-omics Pathways. *Computer Graphics Forum* 42, 3 (June 2023), 259–270. <https://doi.org/10.1111/cgf.14828>
- [36] N. Brich, C. Schulz, J. Peter, W. Klingert, M. Schenk, D. Weiskopf, and M. Krone. 2022. Visual Analytics of Multivariate Intensive Care Time Series Data. *Computer Graphics Forum* 41, 6 (September 2022), 273–286. <https://doi.org/10.1111/cgf.14498>
- [37] Nicolas Brich, Christoph Schulz, Jörg Peter, Wilfried Klingert, Martin Schenk, Daniel Weiskopf, and Michael Krone. 2020. Visual Analysis of Multivariate Intensive Care Surveillance Data. *Eurographics Workshop on Visual Computing for Biology and Medicine* (2020). <https://doi.org/10.2312/VCBM.20201174>
- [38] Carlos Briones. 2023. Polymerase Chain Reaction. *Encyclopedia of Astrobiology*, 2441–2442. https://doi.org/10.1007/978-3-662-65093-6_1252

- [39] John Brooke. 1996. SUS: A 'Quick and Dirty' Usability Scale. *Usability Evaluation In Industry*. <https://doi.org/10.1201/9781498710411>
- [40] Michael Burch, Corinna Vehlow, Natalia Konevtsova, and Daniel Weiskopf. 2012. Evaluating Partially Drawn Links for Directed Graph Edges. *Graph Drawing 7034*, 226–237. https://doi.org/10.1007/978-3-642-25878-7_22
- [41] Atul Butte. 2002. The Use and Analysis of Microarray Data. *Nature Reviews Drug Discovery* 1, 12 (December 2002), 951–960. <https://doi.org/10.1038/nrd961>
- [42] Michael ten Caat, N. M. Maurits, and J. B. T. M. Roerdink. 2005. Tiled Parallel Coordinates for the Visualization of Time-Varying Multichannel EEG Data. In *EUROVIS 2005: Eurographics / IEEE VGTC Symposium on Visualization*, 2005. The Eurographics Association. <https://doi.org/10.2312/VisSym/EuroVis05/061-068>
- [43] Ramon Ferrer i Cancho and Richard V. Solé. 2001. The Small World of Human Language. *Proceedings of the Royal Society of London. Series B: Biological Sciences* 268, 1482 (November 2001), 2261–2265. <https://doi.org/10.1098/rspb.2001.1800>
- [44] M Careri, A Mangia, and M Musci. 1998. Overview of the Applications of Liquid Chromatography–Mass Spectrometry Interfacing Systems in Food Analysis: Naturally Occurring Substances in Food. *Journal of Chromatography A* 794, 1–2 (January 1998), 263–297. [https://doi.org/10.1016/S0021-9673\(97\)00654-7](https://doi.org/10.1016/S0021-9673(97)00654-7)
- [45] Douglas Carroll. 2020. Electrocardiogram (EKG). *Encyclopedia of Behavioral Medicine*, 740–741. https://doi.org/10.1007/978-3-030-39903-0_452
- [46] Eleanor M. Caves, Nicholas C. Brandley, and Sönke Johnsen. 2018. Visual Acuity and the Evolution of Signals. *Trends in Ecology & Evolution* 33, 5 (2018), 358–372. <https://doi.org/10.1016/j.tree.2018.03.001>
- [47] François Chollet and others. 2015. Keras. Retrieved May 3, 2026 from <https://keras.io/>

- [48] Henderson James Cleaves. 2023. Proteins, Primary Structure. *Encyclopedia of Astrobiology*, 2481. https://doi.org/10.1007/978-3-662-65093-6_1271
- [49] Henderson James Cleaves. 2023. Proteins, Secondary Structure. *Encyclopedia of Astrobiology*, 2482. https://doi.org/10.1007/978-3-662-65093-6_1413
- [50] Henderson James Cleaves. 2023. Proteins, Tertiary Structure. *Encyclopedia of Astrobiology*, 2482–2483. https://doi.org/10.1007/978-3-662-65093-6_1573
- [51] Henderson James Cleaves. 2023. Proteins, Quaternary Structure. *Encyclopedia of Astrobiology*, 2481–2482. https://doi.org/10.1007/978-3-662-65093-6_1325
- [52] William S Cleveland and Robert McGill. 1984. Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods. *Journal of the American statistical association* 79, 387 (1984), 531–554. <https://doi.org/10.1080/01621459.1984.10478080>
- [53] Clary B. Clish. 2015. Metabolomics: An Emerging but Powerful Tool for Precision Medicine. *Molecular Case Studies* 1, 1 (October 2015), a588. <https://doi.org/10.1101/mcs.a000588>
- [54] Martin R. Cowie, Juuso I. Blomster, Lesley H. Curtis, Sylvie Duclaux, Ian Ford, Fleur Fritz, Samantha Goldman, Salim Janmohamed, Jörg Kreuzer, Mark Leenay, Alexander Michel, Seleen Ong, Jill P. Pell, Mary Ross Southworth, Wendy Gattis Stough, Martin Thoenes, Faiez Zannad, and Andrew Zalewski. 2017. Electronic Health Records to Facilitate Clinical Research. *Clinical Research in Cardiology* 106, 1 (January 2017), 1–9. <https://doi.org/10.1007/s00392-016-1025-6>
- [55] F. H. Crick. 1958. On Protein Synthesis. *Symposia of the Society for Experimental Biology* 12, (1958), 138–163.
- [56] Lindsey C. Crosswell and Janet M. Thornton. 2012. ELIXIR: A Distributed Infrastructure for European Biological Data. *Trends in*

- Biotechnology* 30, 5 (May 2012), 241–242. <https://doi.org/10.1016/j.tibtech.2012.02.002>
- [57] António Cruz, Joel P Arrais, and Penousal Machado. 2019. Interactive and Coordinated Visualization Approaches for Biological Data Analysis. *Briefings in Bioinformatics* 20, 4 (July 2019), 1513–1523. <https://doi.org/10.1093/bib/bby019>
- [58] J.A. Cuesta-Albertos and A. Nieto-Reyes. 2008. The Random Tukey Depth. *Comput. Stat. Data Anal.* 52, 11 (2008), 4979–4988. <https://doi.org/10.1016/j.csda.2008.04.021>
- [59] G. Dale and A.L. Latner. 1969. Isoelectric Focusing of Serum Proteins in Acrylamide Gels Followed by Electrophoresis. *Clinica Chimica Acta* 24, 1 (April 1969), 61–68. [https://doi.org/10.1016/0009-8981\(69\)90141-7](https://doi.org/10.1016/0009-8981(69)90141-7)
- [60] Tommy Dang, Paul Murray, and Angus Forbes. 2017. BioLinker: Bottom-up Exploration of Protein Interaction Networks. In *2017 IEEE Pacific Visualization Symposium (PacificVis)*, April 2017. 265–269. <https://doi.org/10/gm5wzh>
- [61] C. Scott Dempwolf and Ben Shneiderman. 2017. Event Analytics for Innovation Trajectories: Understanding Inputs and Outcomes for Entrepreneurial Success. *Technology & Innovation* 19, 1 (June 2017), 397–413. <https://doi.org/10.21300/19.1.2017.397>
- [62] J. DeRisi, L. Penland, P. O. Brown, M. L. Bittner, P. S. Meltzer, M. Ray, Y. Chen, Y. A. Su, and J. M. Trent. 1996. Use of a cDNA Microarray to Analyse Gene Expression Patterns in Human Cancer. *Nature Genetics* 14, 4 (December 1996), 457–460. <https://doi.org/10.1038/ng1296-457>
- [63] Bruno Domon and Ruedi Aebersold. 2006. Mass Spectrometry and Protein Analysis. *Science* 312, 5771 (April 2006), 212–217. <https://doi.org/10.1126/science.1124619>
- [64] Nadezhda T. Doncheva, John H. Morris, Jan Gorodkin, and Lars J. Jensen. 2019. Cytoscape StringApp: Network Analysis and Visualization of Proteomics Data. *Journal of Proteome Research* 18, 2

- (February 2019), 623–632. <https://doi.org/10.1021/acs.jproteome.8b00702>
- [65] Geoffrey M. Draper, Yarden Livnat, and Richard F. Riesenfeld. 2009. A Survey of Radial Methods for Information Visualization. *IEEE Transactions on Visualization and Computer Graphics* 15, 5 (September 2009), 759–776. <https://doi.org/10.1109/TVCG.2009.23>
 - [66] Cody Dunne and Ben Shneiderman. 2013. Motif Simplification: Improving Network Visualization Readability with Fan, Connector, and Clique Glyphs. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, April 2013. ACM, Paris France, 3247–3256. <https://doi.org/10.1145/2470654.2466444>
 - [67] J S Dussaut, C A Gallo, F Cravero, M J Martínez, J A Carballido, and I Ponzoni. 2017. GeRNet: A Gene Regulatory Network Tool. *Biosystems*. 162, (December 2017), 1–11. <https://doi.org/10/gnhwwt>
 - [68] Marian Dörk, Rob Comber, and Martyn Dade-Robertson. 2014. Monadic Exploration: Seeing the Whole through Its Parts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, April 2014. ACM, Toronto Ontario Canada, 1535–1544. <https://doi.org/10.1145/2556288.2557083>
 - [69] Valeria D’Argenio. 2018. The High-Throughput Analyses Era: Are We Ready for the Data Struggle?. *High-Throughput* 7, 1 (March 2018), 8. <https://doi.org/10.3390/ht7010008>
 - [70] Henry Ehlers, Nicolas Brich, Michael Krone, Martin Nöllenburg, Jiacheng Yu, Hiroaki Natsukawa, Xiaoru Yuan, and Hsiang-Yun Wu. 2025. An Introduction to and Survey of Biological Network Visualization. *Computers & Graphics* 126, (February 2025), 104115. <https://doi.org/10.1016/j.cag.2024.104115>
 - [71] John Eid, Adrian Fehr, Jeremy Gray, Khai Luong, John Lyle, Geoff Otto, Paul Peluso, David Rank, Primo Baybayan, Brad Bettman, Arkadiusz Bibillo, Keith Bjornson, Bidhan Chaudhuri, Frederick Christians, Ronald Cicero, Sonya Clark, Ravindra Dalal, Alex deWinter, John Dixon, Mathieu Foquet, Alfred Gaertner, Paul

- Hardenbol, Cheryl Heiner, Kevin Hester, David Holden, Gregory Kearns, Xiangxu Kong, Ronald Kuse, Yves Lacroix, Steven Lin, Paul Lundquist, Congcong Ma, Patrick Marks, Mark Maxham, Devon Murphy, Insil Park, Thang Pham, Michael Phillips, Joy Roy, Robert Sebra, Gene Shen, Jon Sorenson, Austin Tomaney, Kevin Travers, Mark Trulson, John Vieceli, Jeffrey Wegener, Dawn Wu, Alicia Yang, Denis Zaccarin, Peter Zhao, Frank Zhong, Jonas Korlach, and Stephen Turner. 2009. Real-Time DNA Sequencing from Single Polymerase Molecules. *Science* 323, 5910 (January 2009), 133–138. <https://doi.org/10.1126/science.1162986>
- [72] Jordan M. Eizenga, Adam M. Novak, Jonas A. Sibbesen, Simon Heumos, Ali Ghaffaari, Glenn Hickey, Xian Chang, Josiah D. Seaman, Robin Rounthwaite, Jana Ebler, Mikko Rautiainen, Shilpa Garg, Benedict Paten, Tobias Marschall, Jouni Sirén, and Erik Garrison. 2020. Pangenome Graphs. *Annual Review of Genomics and Human Genetics* 21, 1 (2020), 139–162. <https://doi.org/10/ghtrkn>
- [73] Stef Van Den Elzen and Jarke J. Van Wijk. 2014. Multivariate Network Exploration and Presentation: From Detail to Overview via Selections and Aggregations. *IEEE Transactions on Visualization and Computer Graphics* 20, 12 (December 2014), 2310–2319. <https://doi.org/10.1109/TVCG.2014.2346441>
- [74] Stef van den Elzen, Danny Holten, Jorik Blaas, and Jarke J. van Wijk. 2016. Reducing Snapshots to Points: A Visual Analytics Approach to Dynamic Network Exploration. *IEEE Trans. Vis. Comput. Graphics* 22, 1 (2016), 1–10. <https://doi.org/10.1109/TVCG.2015.2468078>
- [75] Abdul-Hamid Emwas, Raja Roy, Ryan T. McKay, Leonardo Tenori, Edoardo Saccenti, G. A. Nagana Gowda, Daniel Raftery, Fatimah Alahmari, Lukasz Jaremko, Mariusz Jaremko, and David S. Wishart. 2019. NMR Spectroscopy for Metabolomics Research. *Metabolites* 9, 7 (June 2019), 123. <https://doi.org/10.3390/metabo9070123>

- [76] K. Enge, E. Elmquist, V. Caiola, N. Rönnerberg, A. Rind, M. Iber, S. Lenzi, F. Lan, R. Höldrich, and W. Aigner. 2024. Open Your Ears and Take a Look: A State-of-the-Art Report on the Integration of Sonification and Visualization. *Computer Graphics Forum* 43, 3 (2024), e15114. <https://doi.org/10.1111/cgf.15114>
- [77] Antonio Fabregat, Steven Jupe, Lisa Matthews, Konstantinos Sidiropoulos, Marc Gillespie, Phani Garapati, Robin Haw, Bijay Jassal, Florian Korninger, Bruce May, Marija Milacic, Corina Dueñas Roca, Karen Rothfels, Cristoffer Sevilla, Veronica Shamovsky, Solomon Shorser, Thawfeek Varusai, Guilherme Viteri, Joel Weiser, Guanming Wu, Lincoln Stein, Henning Hermjakob, and Peter D'Eustachio. 2018. The Reactome Pathway Knowledgebase. *Nucleic Acids Research* 46, D1 (January 2018), D649–D655. <https://doi.org/10.1093/nar/gkx1132>
- [78] Facebook. 2013. React - A JavaScript Library for Building User Interfaces. Retrieved May 3, 2026 from <https://reactjs.org/>
- [79] Michael Farrugia, Neil Hurley, and Aaron Quigley. 2011. Exploring Temporal Ego Networks Using Small Multiples and Tree-ring Layouts. In *ACHI 2011: The Fourth International Conference on Advances in Computer-Human Interactions*, 2011. IARIA, Guadeloupe, France, 79–88.
- [80] Joanne E. Fehr, Gayle W. Trotter, Julia T. Oxford, and David A. Hart. 2000. Comparison of Northern Blot Hybridization and a Reverse Transcriptase-Polymerase Chain Reaction Technique for Measurement of mRNA Expression of Metalloproteinases and Matrix Components in Articular Cartilage and Synovial Membrane from Horses with Osteoarthritis. *American Journal of Veterinary Research* 61, 8 (August 2000), 900–905. <https://doi.org/10.2460/ajvr.2000.61.900>
- [81] Stefan P. Feyer, Bruno Pinaud, Stephen Kobourov, Nicolas Brich, Michael Krone, Andreas Kerren, Michael Behrisch, Falk Schreiber, and Karsten Klein. 2024. 2D, 2.5D, or 3D? An Exploratory Study on Multilayer Network Visualisations in Virtual Reality. *IEEE Trans-*

- actions on Visualization and Computer Graphics* 30, 1 (January 2024), 469–479. <https://doi.org/10.1109/TVCG.2023.3327402>
- [82] Oliver Fiehn. 2016. Metabolomics by Gas Chromatography–Mass Spectrometry: Combined Targeted and Untargeted Profiling. *Current Protocols in Molecular Biology* 114, 1 (April 2016). <https://doi.org/10.1002/0471142727.mb3004s114>
- [83] Dustin A. Fife and Joseph Lee Rodgers. 2022. Understanding the Exploratory/Confirmatory Data Analysis Continuum: Moving beyond the “Replication Crisis”. *American Psychologist* 77, 3 (April 2022), 453–466. <https://doi.org/10.1037/amp0000886>
- [84] Velitchko Filipov, Alessio Arleo, and Silvia Miksch. 2023. Are We There Yet? A Roadmap of Network Visualization from Surveys to Task Taxonomies. *Computer Graphics Forum* 42, 6 (September 2023), e14794. <https://doi.org/10.1111/cgf.14794>
- [85] Fabian Fischer, Johannes Fuchs, and Florian Mansmann. 2012. ClockMap: Enhancing Circular Treemaps with Temporal Glyphs for Time-Series Data. In *EuroVis - Short Papers*, 2012. The Eurographics Association. <https://doi.org/10.2312/PE/EUROVISSHORT/EUROVISSHORT2012/097-101>
- [86] James H. Fowler and Nicholas A. Christakis. 2008. Dynamic Spread of Happiness in a Large Social Network: Longitudinal Analysis over 20 Years in the Framingham Heart Study. *BMJ* 337, dec42 (December 2008), a2338–a2338. <https://doi.org/10.1136/bmj.a2338>
- [87] Bruce J. Frayman and William E. Dawson. 1981. The Effect of Object Shape and Mode of Presentation on Judgments of Apparent Volume. *Perception & Psychophysics* 29, 1 (January 1981), 56–62. <https://doi.org/10.3758/BF03198840>
- [88] Kun Fu, Tingyun Mao, Yang Wang, Daoyu Lin, Yuanben Zhang, and Xian Sun. 2021. DyEgoVis: Visual Exploration of Dynamic Ego-Network Evolution. *Applied Sciences* 11, 5 (March 2021), 2399. <https://doi.org/10.3390/app11052399>

- [89] Takanori Fujiwara, Jia-Kai Chou, Shilpika Shilpika, Panpan Xu, Liu Ren, and Kwan-Liu Ma. 2020. An Incremental Dimensionality Reduction Method for Visualizing Streaming Multidimensional Data. *IEEE Trans. Vis. Comput. Graphics* 26, 1 (2020), 418–428. <https://doi.org/10.1109/TVCG.2019.2934433>
- [90] Ambrose Furey, Merisa Moriarty, Vaishali Bane, Brian Kinsella, and Mary Lehane. 2013. Ion Suppression; A Critical Review on Causes, Evaluation, Prevention and Applications. *Talanta* 115, (October 2013), 104–122. <https://doi.org/10.1016/j.talanta.2013.03.048>
- [91] David E. Garfin. 2003. Two-Dimensional Gel Electrophoresis: An Overview. *TrAC Trends in Analytical Chemistry* 22, 5 (May 2003), 263–272. [https://doi.org/10.1016/S0165-9936\(03\)00506-5](https://doi.org/10.1016/S0165-9936(03)00506-5)
- [92] Ellen J Gerl and Molly R Morris. 2008. The Causes and Consequences of Color Vision. *Evolution: Education and Outreach* 1, (2008), 476–486. <https://doi.org/10.1007/s12052-008-0088-x>
- [93] Marc Gillespie, Bijay Jassal, Ralf Stephan, Marija Milacic, Karen Rothfels, Andrea Senff-Ribeiro, Johannes Griss, Cristoffer Sevilla, Lisa Matthews, Chuqiao Gong, Chuan Deng, Thawfeek Varusai, Eliot Ragueneau, Yusra Haider, Bruce May, Veronica Shamovsky, Joel Weiser, Timothy Brunson, Nasim Sanati, Liam Beckman, Xiang Shao, Antonio Fabregat, Konstantinos Sidiropoulos, Julieth Murillo, Guilherme Viteri, Justin Cook, Solomon Shorser, Gary Bader, Emek Demir, Chris Sander, Robin Haw, Guanming Wu, Lincoln Stein, Henning Hermjakob, and Peter D’Eustachio. 2022. The Reactome Pathway Knowledgebase 2022. *Nucleic Acids Research* 50, D1 (January 2022), D687–D692. <https://doi.org/10.1093/nar/gkab1028>
- [94] Prerna Giri and Bhagyalaxmi Mohapatra. 2022. Copy Number Variant (CNV). *Encyclopedia of Animal Cognition and Behavior*, 1714–1717. https://doi.org/10.1007/978-3-319-55065-7_8
- [95] Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George

- B Moody, Chung-Kang Peng, and H Eugene Stanley. 2000. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. *Circulation* 101, 23 (2000), e215–e220. <https://doi.org/10.1161/01.CIR.101.23.e215>
- [96] Mary J. Goldman, Brian Craft, Mim Hastie, Kristupas Repečka, Fran McDade, Akhil Kamath, Ayan Banerjee, Yunhai Luo, Dave Rogers, Angela N. Brooks, Jingchun Zhu, and David Haussler. 2020. Visualizing and Interpreting Cancer Genomics Data via the Xena Platform. *Nature Biotechnology* 38, 6 (June 2020), 675–678. <https://doi.org/10.1038/s41587-020-0546-8>
- [97] Stanley J. Goldsmith. 1975. Radioimmunoassay: Review of Basic Principles. *Seminars in Nuclear Medicine* 5, 2 (April 1975), 125–152. [https://doi.org/10.1016/S0001-2998\(75\)80028-6](https://doi.org/10.1016/S0001-2998(75)80028-6)
- [98] Emanuel Gonçalves, Rebecca C. Poulos, Zhaoxiang Cai, Syd Barthorpe, Srikanth S. Manda, Natasha Lucas, Alexandra Beck, Daniel Bucio-Noble, Michael Dausmann, Caitlin Hall, Michael Hecker, Jennifer Koh, Howard Lightfoot, Sadia Mahboob, Iman Mali, James Morris, Laura Richardson, Akila J. Seneviratne, Rebecca Shepherd, Erin Sykes, Frances Thomas, Sara Valentini, Steven G. Williams, Yangxiu Wu, Dylan Xavier, Karen L. MacKenzie, Peter G. Hains, Brett Tully, Phillip J. Robinson, Qing Zhong, Mathew J. Garnett, and Roger R. Reddel. 2022. Pan-Cancer Proteomic Map of 949 Human Cell Lines. *Cancer Cell* 40, 8 (August 2022), 835–849.e8. <https://doi.org/10.1016/j.ccell.2022.06.010>
- [99] Miriam Greis, Emre Avci, Albrecht Schmidt, and Tonja Machulla. 2017. Increasing Users' Confidence in Uncertain Data by Aggregating Data from Multiple Sources. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*, 2017. Association for Computing Machinery, New York, NY, USA, 828–840. <https://doi.org/10.1145/3025453.3025998>
- [100] S. Grottel, J. Heinrich, D. Weiskopf, and S. Gumhold. 2014. Visual Analysis of Trajectories in Multi-Dimensional State Spaces. *Com-*

- puter Graphics Forum* 33, 6 (September 2014), 310–321. <https://doi.org/10.1111/cgf.12352>
- [101] Andrea Guarracino, Simon Heumos, Sven Nahnsen, Pjotr Prins, and Erik Garrison. 2022. ODGI: Understanding Pangenome Graphs. *bioRxiv : the preprint server for biology* (2022). <https://doi.org/10.1101/2021.11.10.467921>
 - [102] Arun Gupta. 2024. Sick Cell Anemia and Related Hemoglobinopathies. In *Decision Making Through Problem Based Learning in Hematology*. Springer Nature Singapore, Singapore, 269–289. https://doi.org/10.1007/978-981-99-8933-1_21
 - [103] James Hadfield, Colin Megill, Sidney M Bell, John Huddleston, Barney Potter, Charlton Callender, Pavel Sagulenko, Trevor Bedford, and Richard A Neher. 2018. Nextstrain: Real-Time Tracking of Pathogen Evolution. *Bioinformatics* 34, 23 (December 2018), 4121–4123. <https://doi.org/10.1093/bioinformatics/bty407>
 - [104] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. 2008. Exploring Network Structure, Dynamics, and Function Using NetworkX. In *Proceedings of the 7th Python in Science Conference*, 2008. Pasadena, CA USA, 11–15.
 - [105] Frank van Ham and Adam Perer. 2009. "Search, Show Context, Expand on Demand": Supporting Large Graph Exploration with Degree-of-Interest. *IEEE Transactions on Visualization and Computer Graphics* 15, 6 (November 2009), 953–960. <https://doi.org/10.1109/TVCG.2009.108>
 - [106] Xuemei Han, Aaron Aslanian, and John R Yates. 2008. Mass Spectrometry for Proteomics. *Current Opinion in Chemical Biology* 12, 5 (October 2008), 483–490. <https://doi.org/10.1016/j.cbpa.2008.07.024>
 - [107] Theresa Harbig. 2025. From Microbes to Medical Cohorts: Visualizing Multi-Omics Data on Different Scales. <https://doi.org/10.15496/PUBLIKATION-109403>
 - [108] S. Haroz, R. Kosara, and S. Franconeri. 2016. The Connected Scatterplot for Presenting Paired Time Series. *IEEE Trans. Vis. Comput.*

- Graphics* 22, 9 (September 2016), 2174–2186. <https://doi.org/10.1109/TVCG.2015.2502587>
- [109] Mark Harrower and Cynthia A Brewer. 2003. ColorBrewer. Org: An Online Tool for Selecting Colour Schemes for Maps. *The Cartographic Journal* 40, 1 (2003), 27–37. <https://doi.org/10.1179/000870403235002042>
- [110] Sheng He, Yi-Jun Liu, Fei-Yue Ye, Ren-Pu Li, and Ren-Jun Dai. 2019. A New Grid- and Modularity-Based Layout Algorithm for Complex Biological Networks. *PLOS ONE* 14, 8 (August 2019), e221620. <https://doi.org/10.1371/journal.pone.0221620>
- [111] Jeffrey Heer and Michael Bostock. 2010. Crowdsourcing Graphical Perception: Using Mechanical Turk to Assess Visualization Design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Chi '10)*, 2010. Association for Computing Machinery, Atlanta, Georgia, USA and New York, NY, USA, 203–212. <https://doi.org/10.1145/1753326.1753357>
- [112] Ginny Hendricks, Dominika Tkaczyk, Jennifer Lin, and Patricia Feeney. 2020. Crossref: The Sustainable Source of Community-Owned Scholarly Metadata. *Quantitative Science Studies* 1, 1 (February 2020), 414–427. https://doi.org/10.1162/qss_a_00022
- [113] Nathalie Henry, Jean-Daniel Fekete, and Michael J. McGuffin. 2007. NodeTrix: A Hybrid Visualization of Social Networks. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (November 2007), 1302–1309. <https://doi.org/10.1109/TVCG.2007.70582>
- [114] N. Heulot, M. Aupetit, and J-D. Fekete. 2013. ProxiLens: Interactive Exploration of High-Dimensional Data Using Projections. In *EuroVis Workshop on Visual Analytics Using Multidimensional Projections*, 2013. The Eurographics Association. <https://doi.org/10.2312/PE.VAMP.VAMP2013.011-015>
- [115] Russell Higuchi, Gavin Dollinger, P. Sean Walsh, and Robert Griffith. 1992. Simultaneous Amplification and Detection of Spe-

- cific DNA Sequences. *Bio/Technology* 10, 4 (April 1992), 413–417. <https://doi.org/10.1038/nbt0492-413>
- [116] Alexander Hinneburg and Daniel A Keim. 1999. Optimal Grid-Clustering : Towards Breaking the Curse of Dimensionality in High-Dimensional Clustering. In *VLDB '99: Proceedings of the 25th International Conference on Very Large Data Bases*, 1999. Morgan Kaufmann Publishers Inc., San Francisco, CA, 506–517.
- [117] Andreas Hinterreiter, Christian Steinparz, Moritz Schöfl, Holger Stitz, and Marc Streit. 2021. Projection Path Explorer: Exploring Visual Patterns in Projected Decision-Making Paths. *ACM Transactions on Interactive Intelligent Systems* 11, 3–4 (August 2021), 1–29. <https://doi.org/10.1145/3387165>
- [118] Miriam Hoene, Lisa Kappler, Laxmikanth Kollipara, Chunxiu Hu, Martin Irmeler, Daniel Bleher, Christoph Hoffmann, Johannes Beckers, Martin Hrabě de Angelis, Hans-Ulrich Häring, Andreas L. Birkenfeld, Andreas Peter, Albert Sickmann, Guowang Xu, Rainer Lehmann, and Cora Weigert. 2021. Exercise Prevents Fatty Liver by Modifying the Compensatory Response of Mitochondrial Metabolism to Excess Substrate Availability. *Molecular Metabolism* 54, (December 2021), 101359. <https://doi.org/10.1016/j.molmet.2021.101359>
- [119] Danny Holten. 2006. Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data. *IEEE Transactions on Visualization and Computer Graphics* 12, 5 (September 2006), 741–748. <https://doi.org/10.1109/TVCG.2006.147>
- [120] Harold Hotelling. 1933. Analysis of a Complex of Statistical Variables into Principal Components. 24, 6 (1933), 417. <https://doi.org/10.1037/h0071325>
- [121] Petra Isenberg, Florian Heimerl, Steffen Koch, Tobias Isenberg, Panpan Xu, Charles D. Stolper, Michael Sedlmair, Jian Chen, Torsten Moller, and John Stasko. 2017. Vispubdata.Org: A Metadata Collection About IEEE Visualization (VIS) Publications. *IEEE Transactions on Visualization and Computer Graphics* 23, 9

- (September 2017), 2199–2206. <https://doi.org/10.1109/TVCG.2016.2615308>
- [122] D. Jackle, F. Fischer, T. Schreck, and D. Keim. 2016. Temporal MDS Plots for Analysis of Multivariate Data. *IEEE Trans. Vis. Comput. Graphics* 22, 1 (2016), 141–150. <https://doi.org/10.1109/TVCG.2015.2467553>
 - [123] Mathieu Jacomy, Tommaso Venturini, Sebastien Heymann, and Mathieu Bastian. 2014. ForceAtlas2, a Continuous Graph Layout Algorithm for Handy Network Visualization Designed for the Gephi Software. *PLOS ONE* 9, 6 (June 2014), e98679. <https://doi.org/10/f3mn4z>
 - [124] Yvonne Jansen, Pierre Dragicevic, Petra Isenberg, Jason Alexander, Abhijit Karnik, Johan Kildal, Sriram Subramanian, and Kasper Hornbæk. 2015. Opportunities and Challenges for Data Physicalization. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, April 2015. ACM, Seoul Republic of Korea, 3227–3236. <https://doi.org/10.1145/2702123.2702180>
 - [125] Alistair E. W. Johnson, Tom J. Pollard, Lu Shen, Li-wei H. Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. 2016. MIMIC-III, a Freely Accessible Critical Care Database. *Scientific Data* 3, 1 (2016), 1–9. <https://doi.org/10.1038/sdata.2016.35>
 - [126] Julia H. Joo, Sujata Rao, and Arun D. Singh. 2025. Cancer Angiogenesis. *Clinical Ophthalmic Oncology: Basic Principles*, 51–71. https://doi.org/10.1007/978-3-031-75907-9_6
 - [127] G. Joshi-Tope. 2004. Reactome: A Knowledgebase of Biological Pathways. *Nucleic Acids Research* 33, Database issue (December 2004), D428–D432. <https://doi.org/10.1093/nar/gki072>
 - [128] Björn H Junker, Christian Klukas, and Falk Schreiber. 2006. VANTED: A System for Advanced Data Analysis and Visualization in the Context of Biological Networks. *BMC Bioinformatics* 7, 1 (December 2006), 109. <https://doi.org/10.1186/1471-2105-7-109>

- [129] Bharat Kale, Maoyuan Sun, and Michael E. Papka. 2023. The State of the Art in Visualizing Dynamic Multivariate Networks. *Computer Graphics Forum* 42, 3 (June 2023), 471–490. <https://doi.org/10.1111/cgf.14856>
- [130] Minoru Kanehisa and Susumu Goto. 2000. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research* 28, 1 (January 2000), 27–30. Retrieved January 31, 2022 from <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC102409/>
- [131] Minoru Kanehisa, Michihiro Araki, Susumu Goto, Masahiro Hattori, Mika Hirakawa, Masumi Itoh, Toshihaki Katayama, Shuichi Kawashima, Shujiro Okuda, Toshiaki Tokimatsu, and Yoshihiro Yamanishi. 2007. KEGG for Linking Genomes to Life and the Environment. *Nucleic Acids Research* 36, Database (December 2007), D480–D484. <https://doi.org/10.1093/nar/gkm882>
- [132] Frigyes Karinthy. 1929. Láncszemek. *Minden másképp van* (1929).
- [133] Guy Karlebach and Ron Shamir. 2008. Modelling and Analysis of Gene Regulatory Networks. *Nature Reviews. Molecular Cell Biology* 9, 10 (October 2008), 770–780. <https://doi.org/10.1038/nrm2503>
- [134] Peter D. Karp, Richard Billington, Ron Caspi, Carol A. Fulcher, Mario Latendresse, Anamika Kothari, Ingrid M. Keseler, Markus Krummenacker, Peter E. Midford, Quang Ong, Wai Kit Ong, Suzanne M. Paley, and Pallavi Subhraveti. 2019. The BioCyc Collection of Microbial Genomes and Metabolic Pathways. *Briefings in Bioinformatics* 20, 4 (July 2019), 1085–1093. <https://doi.org/10.1093/bib/bbx085>
- [135] Daishi Kato. 2023. Jotai — Primitive and Flexible State Management for React. Retrieved May 3, 2026 from <https://jotai.org/>
- [136] Johannes Kehrler and Helwig Hauser. 2013. Visualization and Visual Analysis of Multifaceted Scientific Data: A Survey. *IEEE Trans. Vis. Comput. Graphics* 19, 3 (2013), 495–513. <https://doi.org/10.1109/TVCG.2012.110>

- [137] Daniel A. Keim and Daniela Oelke. 2007. Literature Fingerprinting: A New Method for Visual Literary Analysis. In *2007 IEEE Symposium on Visual Analytics Science and Technology*, October 2007. IEEE, Sacramento, CA, USA, 115–122. <https://doi.org/10.1109/VAST.2007.4389004>
- [138] René Keller, Claudia M. Eckert, and P. John Clarkson. 2006. Matrices or Node-Link Diagrams: Which Visual Representation Is Better for Visualising Connectivity Models?. *Information Visualization* 5, 1 (March 2006), 62–76. <https://doi.org/10.1057/palgrave.ivs.9500116>
- [139] Diederik P Kingma and Max Welling. 2013. Auto-Encoding Variational Bayes. <https://doi.org/10.48550/ARXIV.1312.6114>
- [140] Adrienne Kish. 2023. Electrophoresis. *Encyclopedia of Astrobiology*, 886–887. https://doi.org/10.1007/978-3-662-65093-6_500
- [141] Klingert, W., J. Peter, C. Thiel, K. Thiel, W. Rosenstiel, K. Klingert, C Grasshoff, A. Königsrainer, and M. Schenk. 2018. Fully Automated Life Support: An Implementation and Feasibility Pilot Study in Healthy Pigs. *Intensive Care Med. Exp.* 6, 1 (2018). <https://doi.org/10.1186/s40635-018-0168-3>
- [142] Bernhard Korte and Jens Vygen. 2012. *Combinatorial Optimization: Theory and Algorithms*. Springer Berlin Heidelberg, Berlin, Heidelberg. <https://doi.org/10.1007/978-3-642-24488-9>
- [143] Mikaela Koutrouli, Evangelos Karatzas, David Paez-Espino, and Georgios A. Pavlopoulos. 2020. A Guide to Conquer the Biological Network Era Using Graph Theory. *Frontiers in Bioengineering and Biotechnology* 8, (January 2020), 34. <https://doi.org/10.3389/fbioe.2020.00034>
- [144] István A. Kovács, Katja Luck, Kerstin Spirohn, Yang Wang, Carl Pollis, Sadie Schlabach, Wenting Bian, Dae-Kyum Kim, Nishka Kishore, Tong Hao, Michael A. Calderwood, Marc Vidal, and Albert-László Barabási. 2019. Network-Based Prediction of Protein Interactions. *Nature Communications* 10, 1 (March 2019), 1240. <https://doi.org/10.1038/s41467-019-09177-y>

- [145] Barbora Kozlikova, Daniel Jönsson, Renata Raidou, and Sean O'Donoghue. 2023. Bio+MedVis Challenge @ IEEE VIS. Retrieved May 3, 2026 from http://biovis.net/2023/biovisChallenges_vis/
- [146] Per Ola Kristensson, Nils Dahlback, Daniel Anundi, Marius Bjornstad, Hanna Gillberg, Jonas Haraldsson, Ingrid Martensson, Mathias Nordvall, and Josefine Stahl. 2009. An Evaluation of Space Time Cube Representation of Spatiotemporal Patterns. *IEEE Trans. Vis. Comput. Graphics* 15, 4 (2009), 696–702. <https://doi.org/10.1109/TVCG.2008.194>
- [147] Michael Krone, Andreas Dräger, Ebru Cobanoglu, Manuel Otto Harke, Miriam Hoene, Cora Weigert, and Rainer Lehmann. 2020. A Web-based Visual Analytics Application for Biological Networks. *EuroVis 2020 - Posters* (2020). <https://doi.org/10.2312/EURP.20201124>
- [148] Kimberly R. Kukurba and Stephen B. Montgomery. 2015. RNA Sequencing and Analysis. *Cold Spring Harbor Protocols* 2015, 11 (November 2015), pdb.top84970. <https://doi.org/10.1101/pdb.top084970>
- [149] T. von Landesberger, A. Kuijper, T. Schreck, J. Kohlhammer, J.j. van Wijk, J.-D. Fekete, and D.w. Fellner. 2011. Visual Analysis of Large Graphs: State-of-the-Art and Future Research Challenges. *Computer Graphics Forum* 30, 6 (2011), 1719–1749. <https://doi.org/10.1111/j.1467-8659.2011.01898.x>
- [150] Vito Latora and Massimo Marchiori. 2001. Efficient Behavior of Small-World Networks. *Physical Review Letters* 87, 19 (October 2001), 198701. <https://doi.org/10.1103/PhysRevLett.87.198701>
- [151] Franck Lebeau. D3-Voronoi-Map. Retrieved May 3, 2026 from <https://github.com/Kcnarf/d3-voronoi-map>
- [152] Bongshin Lee, Catherine Plaisant, Cynthia Sims Parr, Jean-Daniel Fekete, and Nathalie Henry. 2006. Task Taxonomy for Graph Visualization. In *Proceedings of the 2006 AVI Workshop on BEyond Time and Errors: Novel Evaluation Methods for Information Visual-*

- ization (BELIV '06), May 2006. ACM, Venice Italy, 1–5. <https://doi.org/10.1145/1168149.1168168>
- [153] Jan de Leeuw. 1977. Applications of Convex Analysis to Multidimensional Scaling. *Recent Developments in Statistics*, 133–146.
 - [154] Jingjia Li, Jason B. West, Alexander Hart, Jill L. Wegrzyn, Matthew A. Smith, Jean-Christophe Domec, Carol A. Loopstra, and Claudio Casola. 2021. Extensive Variation in Drought-Induced Gene Expression Changes Between Loblolly Pine Genotypes. *Frontiers in Genetics* 12, (May 2021), 661440. <https://doi.org/10.3389/fgene.2021.661440>
 - [155] Xiaoting Ling, Chenghan Wang, Linlin Li, Liqiu Pan, Chaoyu Huang, Caixia Zhang, Yunhua Huang, Yuling Qiu, Faquan Lin, and Yifang Huang. 2023. Third-Generation Sequencing for Genetic Disease. *Clinica Chimica Acta* 551, (November 2023), 117624. <https://doi.org/10.1016/j.cca.2023.117624>
 - [156] Dongyu Liu, Fangzhou Guo, Bowen Deng, Huamin Qu, and Yingcai Wu. 2017. egoComp: A Node-Link-Based Technique for Visual Comparison of Ego-Networks. *Information Visualization* 16, 3 (July 2017), 179–189. <https://doi.org/10.1177/14738716166667632>
 - [157] Shusen Liu, Dan Maljovec, Bei Wang, Peer-Timo Bremer, and Valerio Pascucci. 2017. Visualizing High-Dimensional Data: Advances in the Past Decade. *IEEE Trans. Vis. Comput. Graphics* 23, 3 (2017), 1249–1268. <https://doi.org/10.1109/TVCG.2016.2640960>
 - [158] Tianyuan Liu, Pedro Salguero, Marko Petek, Carlos Martinez-Mira, Leandro Balzano-Nogueira, Živa Ramšak, Lauren McIntyre, Kristina Gruden, Sonia Tarazona, and Ana Conesa. 2022. PaintOmics 4: New Tools for the Integrative Analysis of Multi-Omics Datasets Supported by Multiple Pathway Databases. *Nucleic Acids Research* 50, W1 (July 2022), W551–W559. <https://doi.org/10.1093/nar/gkac352>
 - [159] Kenneth J. Livak and Thomas D. Schmittgen. 2001. Analysis of Relative Gene Expression Data Using Real-Time Quantitative PCR

- and the 2- $\Delta\Delta$ CT Method. *Methods* 25, 4 (December 2001), 402–408. <https://doi.org/10.1006/meth.2001.1262>
- [160] O. H. Lowry, N. J. Rosebrough, A. L. Farr, and R. J. Randall. 1951. Protein Measurement with the Folin Phenol Reagent. *The Journal of Biological Chemistry* 193, 1 (November 1951), 265–275.
- [161] Weijun Luo, Gaurav Pant, Yeshvant K Bhavnasi, Steven G Blanchard Jr, and Cory Brouwer. 2017. Pathview Web: User Friendly Pathway Visualization and Data Integration. *Nucleic acids research* 45, W1 (2017), W501–W508. <https://doi.org/10.1093/nar/gkx372>
- [162] S. López-Pintado, Y. Sun, J. Lin, and M. Genton. 2014. Simplicial Band Depth for Multivariate Functional Data. *Adv. Data Anal. Classi.* 8, 3 (2014), 321–338. <https://doi.org/10.1007/s11634-014-0166-6>
- [163] Sara López-Pintado and Juan Romo. 2009. On the Concept of Depth for Functional Data. *Journal of the American Statistical Association* 104, 486 (June 2009), 718–734. <https://doi.org/10.1198/jasa.2009.0108>
- [164] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data Using T-SNE. *Journal of Machine Learning Research* 9, 86 (2008), 2579–2605. Retrieved May 3, 2026 from <http://jmlr.org/papers/v9/vandermaaten08a.html>
- [165] Patricia L. Mabry. 2011. Making Sense of the Data Explosion. *American Journal of Preventive Medicine* 40, 5 (May 2011), S159–S161. <https://doi.org/10.1016/j.amepre.2011.02.001>
- [166] Joseph C. Maranville, Francesca Luca, Allison L. Richards, Xiaquan Wen, David B. Witonsky, Shaneen Baxter, Matthew Stephens, and Anna Di Rienzo. 2011. Interactions between Glucocorticoid Treatment and Cis-Regulatory Polymorphisms Contribute to Cellular Response Phenotypes. *PLoS Genetics* 7, 7 (July 2011), e1002162. <https://doi.org/10.1371/journal.pgen.1002162>
- [167] Justin Matejka and George Fitzmaurice. 2017. Same Stats, Different Graphs: Generating Datasets with Varied Appearance and

- Identical Statistics through Simulated Annealing. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, May 2017. ACM, Denver Colorado USA, 1290–1294. <https://doi.org/10.1145/3025453.3025912>
- [168] Runjhun Mathur, Bhisham Singh Rana, and Abhimanyu Kumar Jha. 2022. Single Nucleotide Polymorphism (SNP). *Encyclopedia of Animal Cognition and Behavior*, 6430–6433. https://doi.org/10.1007/978-3-319-55065-7_2049
- [169] L. E. Matzen, B. C. Howell, M. Tuft, and K. M. Divis. 2024. Transparent Risks: The Impact of the Specificity and Visual Encoding of Uncertainty on Decision Making. *Computer Graphics Forum* 43, 3 (June 2024), e15094. <https://doi.org/10.1111/cgf.15094>
- [170] Graeme C. McAlister, W. Travis Berggren, Jens Griep-Raming, Stevan Horning, Alexander Makarov, Doug Phanstiel, George Stafford, Danielle L. Swaney, John E. P. Syka, Vlad Zabrouskov, and Joshua J. Coon. 2008. A Proteomics Grade Electron Transfer Dissociation-Enabled Hybrid Linear Ion Trap-Orbitrap Mass Spectrometer. *Journal of Proteome Research* 7, 8 (August 2008), 3127–3136. <https://doi.org/10.1021/pr800264t>
- [171] Fintan McGee, Mohammad Ghoniem, Guy Melançon, Benoit Otjacques, and Bruno Pinaud. 2019. The State of the Art in Multilayer Network Visualization. *Computer Graphics Forum* 38, 6 (2019), 125–149. <https://doi.org/10.1111/cgf.13610>
- [172] Leland McInnes, John Healy, and James Melville. 2020. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. <https://doi.org/10.48550/arXiv.1802.03426>
- [173] Wes McKinney and others. 2010. Data Structures for Statistical Computing in Python. In *Proc. 9th Python in Science Conference*, 2010. Austin, Texas, USA, 51–56. <https://doi.org/10.25080/Majora-92bf1922-012>
- [174] Christian von Mering, Martijn Huynen, Daniel Jaeggi, Steffen Schmidt, Peer Bork, and Berend Snel. 2003. STRING: A Database of Predicted Functional Associations between Proteins. *Nucleic*

- Acids Research* 31, 1 (2003), 258–261. <https://doi.org/10.1093/nar/gkg034>
- [175] Gerhard Michal. 2017. Roche - Biochemical Pathways. Retrieved February 2, 2024 from <http://biochemical-pathways.com/>
- [176] Stanley Milgram. 1967. The Small World Problem. *Psychology today* 2, 1 (1967), 60–67.
- [177] Charles Joseph Minard. 1869. Carte Figurative Des Pertes Successives En Hommes A de l'armée Qu'Annibal Conduisit d'Espagne En Italie En Traversant Les Gaules (Selon Polybe). B de l'Armée Française Dans La Campagne de Russie 1812-1813.
- [178] Mahsa Mirzargar, Ross T. Whitaker, and Robert M. Kirby. 2014. Curve Boxplot: Generalization of Boxplot for Ensembles of Curves. *IEEE Trans. Vis. Comput. Graphics* 20, 12 (2014), 2654–2663. <https://doi.org/10.1109/TVCG.2014.2346455>
- [179] Megan Monroe, Rongjian Lan, Hanseung Lee, Catherine Plaisant, and Ben Shneiderman. 2013. Temporal Event Sequence Simplification. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (December 2013), 2227–2236. <https://doi.org/10.1109/TVCG.2013.200>
- [180] B. Moody, G. Moody, M. Villarroel, G. Clifford, and I. & Silva. 2020. MIMIC-III Waveform Database Matched Subset (Version 1.0). (2020). <https://doi.org/10.13026/c2294b>
- [181] K. Moreland. 2013. A Survey of Visualization Pipelines. *IEEE Trans. Vis. Comput. Graphics* 19, 3 (2013), 367–378. <https://doi.org/10.1109/TVCG.2012.133>
- [182] The MUI team. 2023. MUI: The React Component Library You Always Wanted. Retrieved May 7, 2024 from <https://mui.com/>
- [183] K. Mullis, F. Faloon, S. Scharf, R. Saiki, G. Horn, and H. Erlich. 1986. Specific Enzymatic Amplification of DNA In Vitro: The Polymerase Chain Reaction. *Cold Spring Harbor Symposia on Quantitative Biology* 51, 0 (January 1986), 263–273. <https://doi.org/10.1101/SQB.1986.051.01.032>

- [184] Tamara Munzner. 2015. *Visualization Analysis and Design*. CRC Press, Taylor & Francis Group, Boca Raton.
- [185] Paul Murray, Fintan McGee, and Angus G Forbes. 2017. A Taxonomy of Visualization Tasks for the Analysis of Biological Pathway Data. *BMC bioinformatics* 18, 2 (2017), 1–13. <https://doi.org/10.1186/s12859-016-1443-5>
- [186] Dale Muzzey, Eric A. Evans, and Caroline Lieber. 2015. Understanding the Basics of NGS: From Mechanism to Variant Calling. *Current Genetic Medicine Reports* 3, 4 (December 2015), 158–165. <https://doi.org/10.1007/s40142-015-0076-8>
- [187] Ugrappa Nagalakshmi, Zhong Wang, Karl Waern, Chong Shou, Debasish Raha, Mark Gerstein, and Michael Snyder. 2008. The Transcriptional Landscape of the Yeast Genome Defined by RNA Sequencing. *Science* 320, 5881 (June 2008), 1344–1349. <https://doi.org/10.1126/science.1158441>
- [188] Shinichi Nakagawa, David W. Armitage, Tom Froese, Yefeng Yang, and Malgorzata Lagisz. 2025. Poor Hypotheses and Research Waste in Biology: Learning from a Theory Crisis in Psychology. *BMC Biology* 23, 1 (February 2025), 33. <https://doi.org/10.1186/s12915-025-02134-w>
- [189] Tamás Nepusz, Haiyuan Yu, and Alberto Paccanaro. 2012. Detecting Overlapping Protein Complexes in Protein-Protein Interaction Networks. *Nature Methods* 9, 5 (March 2012), 471–472. <https://doi.org/10.1038/nmeth.1938>
- [190] Mark Newman, Albert-László Barabási, and Duncan J. Watts. 2006. *The Structure and Dynamics of Networks*. Princeton University Press. <https://doi.org/10.1515/9781400841356>
- [191] Florence Nightingale, William Farr, and Andrew Smith. 1859. *A Contribution to the Sanitary History of the British Army during the Late War with Russia*. John W. Parker and Son, London.
- [192] Carolina Nobre, Miriah Meyer, Marc Streit, and Alexander Lex. 2019. The State of the Art in Visualizing Multivariate Networks.

- Computer Graphics Forum* 38, 3 (June 2019), 807–832. <https://doi.org/10.1111/cgf.13728>
- [193] Arlind Nocaj and Ulrik Brandes. 2012. Computing Voronoi Treemaps: Faster, Simpler, and Resolution-independent. *Computer Graphics Forum* 31, 3pt1 (June 2012), 855–864. <https://doi.org/10.1111/j.1467-8659.2012.03078.x>
- [194] Arlind Nocaj, Mark Ortmann, and Ulrik Brandes. 2014. Untangling Hairballs: From 3 to 14 Degrees of Separation. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications (Lecture Notes in Computer Science)*, 2014. Springer International Publishing, Cham, 101–112. https://doi.org/10.1007/978-3-662-45803-7_9
- [195] Alberto Noronha, Anna Dröfn Daniélsdóttir, Piotr Gawron, Freyr Jóhannsson, Soffia Jónsdóttir, Sindri Jarlsson, Jón Pétur Gunnarsson, Sigurður Brynjólfsson, Reinhard Schneider, Ines Thiele, and Ronan M. T. Fleming. 2017. ReconMap: An Interactive Visualization of Human Metabolism. *Bioinformatics (Oxford, England)* 33, 4 (February 2017), 605–607. <https://doi.org/10.1093/bioinformatics/btw667>
- [196] S. Nusrat, T. Harbig, and N. Gehlenborg. 2019. Tasks, Techniques, and Tools for Genomic Data Visualization. *Computer Graphics Forum* 38, 3 (June 2019), 781–805. <https://doi.org/10.1111/cgf.13727>
- [197] Shao-En Ong and Akhilesh Pandey. 2001. An Evaluation of the Use of Two-Dimensional Gel Electrophoresis in Proteomics. *Biomolecular Engineering* 18, 5 (November 2001), 195–205. [https://doi.org/10.1016/S1389-0344\(01\)00095-8](https://doi.org/10.1016/S1389-0344(01)00095-8)
- [198] Patricia Ordóñez, Marie desJardins, Michael Lombardi, Christoph U. Lehmann, and Jim Fackler. 2010. An Animated Multivariate Visualization for Physiological and Clinical Data in the ICU. In *Proceedings of the 1st ACM International Health Informatics Symposium (IHI '10)*, 2010. Association for Computing Machinery, New York, NY, USA, 771–779. <https://doi.org/10.1145/1882992.1883109>

- [199] D. Pahr, S. Di Bartolomeo, H. Ehlers, V. A. Filipov, C. Stoiber, W. Aigner, H.-Y. Wu, and R. G. Raidou. 2025. NODKANT: Exploring Constructive Network Physicalization. *Computer Graphics Forum* 44, 3 (June 2025), e70140. <https://doi.org/10.1111/cgf.70140>
- [200] Subhajit Pal, Sudip Mondal, Gourab Das, Sunirmal Khatua, and Zhumur Ghosh. 2020. Big Data in Biology: The Hope and Present-Day Challenges in It. *Gene Reports* 21, (December 2020), 100869. <https://doi.org/10.1016/j.genrep.2020.100869>
- [201] Sabrina Paneels and Jonathan C Roberts. 2010. Review of Designs for Haptic Data Visualization. *IEEE Transactions on Haptics* 3, 2 (April 2010), 119–137. <https://doi.org/10.1109/TOH.2009.44>
- [202] Zhiqiang Pang, Jasmine Chong, Guangyan Zhou, David Anderson de~Lima~Morais, Le Chang, Michel Barrette, Carol Gauthier, Pierre-Étienne Jacques, Shuzhao Li, and Jianguo Xia. 2021. MetaboAnalyst 5.0: Narrowing the Gap between Raw Spectra and Functional Insights. *Nucleic Acids Research* 49, W1 (July 2021), W388–W396. <https://doi.org/10.1093/nar/gkab382>
- [203] Ab Pant. 2024. Metabolomics. *Dictionary of Toxicology*, 613. https://doi.org/10.1007/978-981-99-9283-6_1675
- [204] Ab Pant. 2024. MicroRNA. *Dictionary of Toxicology*, 642. https://doi.org/10.1007/978-981-99-9283-6_1736
- [205] Louis Papageorgiou, Picasi Eleni, Sofia Raftopoulou, Meropi Mantaïou, Vasileios Megalooikonomou, and Dimitrios Vlachakis. 2018. Genomic Big Data Hitting the Storage Bottleneck. *EMBnet.journal* 24, (2018), e910.
- [206] Karl Pearson. 1901. LIII. On Lines and Planes of Closest Fit to Systems of Points in Space. *Lond. Edinb. Dublin Philos. Mag. J. Sci.* 2, 11 (1901), 559–572. <https://doi.org/10.1080/14786440109462720>
- [207] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-Learn: Ma-

- chine Learning in Python. *Journal of Machine Learning Research* 12, 85 (2011), 2825–2830. Retrieved May 7, 2024 from <http://jmlr.org/papers/v12/pedregosa11a.html>
- [208] Michele Pellizzari. 2010. Do Friends and Relatives Really Help in Getting a Good Job?. *ILR Review* 63, 3 (April 2010), 494–510. <https://doi.org/10.1177/001979391006300307>
- [209] Adam Perer and Ben Shneiderman. 2008. Integrating Statistics and Visualization: Case Studies of Gaining Clarity during Exploratory Data Analysis. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, April 2008. ACM, Florence Italy, 265–274. <https://doi.org/10.1145/1357054.1357101>
- [210] Luiz Pessoa. 2014. Understanding Brain Networks and Brain Organization. *Physics of Life Reviews* 11, 3 (September 2014), 400–435. <https://doi.org/10.1016/j.plrev.2014.03.005>
- [211] Jörg Peter, Wilfried Klingert, Alfred Königsrainer, Wolfgang Rosenstiel, Martin Bogdan, and Martin Schenk. 2014. TICoMS – A Modular and Message-Based Framework for Monitoring and Control of Medical Devices. In *IEEE 27th International Symposium on Computer-Based Medical Systems*, 2014. IEEE, New York, NY, USA, 473–474. <https://doi.org/10.1109/CBMS.2014.96>
- [212] Jörg Peter. 2018. Assessment of Goal-Directed Closed-Loop Management in Intensive Care Medicine. Doctoral dissertation. <https://doi.org/10.15496/PUBLIKATION-25018>
- [213] Catherine Plaisant, Brett Milash, Anne Rose, Seth Widoff, and Ben Shneiderman. 1996. LifeLines: Visualizing Personal Histories. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems Common Ground - CHI '96*, 1996. ACM Press, Vancouver, British Columbia, Canada, 221–ff. <https://doi.org/10.1145/238386.238493>
- [214] W. Playfair. 1786. *The Commercial and Political Atlas*. J. Debrett, London.

- [215] Guillaume Plique and Alexis Jacomy. Sigma.js: A JavaScript Library Aimed at Visualizing Graphs of Thousands of Nodes and Edges. Retrieved February 2, 2024 from <https://www.sigmajs.org/>
- [216] Jason Portenoy, Jessica Hullman, and Jevin D. West. 2017. Leveraging Citation Networks to Visualize Scholarly Influence Over Time. *Frontiers in Research Metrics and Analytics* 2, (November 2017), 8. <https://doi.org/10.3389/frma.2017.00008>
- [217] Morgan N. Price, Paramvir S. Dehal, and Adam P. Arkin. 2010. FastTree 2 – Approximately Maximum-Likelihood Trees for Large Alignments. *PLOS ONE* 5, 3 (March 2010), e9490. <https://doi.org/10.1371/journal.pone.0009490>
- [218] Joseph Priestley. 1769. *A New Chart of History*. Engraved and published... by J. Johnson in Pater-noster Row.
- [219] Qingsong Liu, Yifan Hu, Lei Shi, Xinzhu Mu, Yutao Zhang, and Jie Tang. 2015. EgoNetCloud: Event-based Egocentric Dynamic Network Visualization. In *2015 IEEE Conference on Visual Analytics Science and Technology (VAST)*, October 2015. IEEE, Chicago, IL, USA, 65–72. <https://doi.org/10.1109/VAST.2015.7347632>
- [220] Michael Quail, Miriam E Smith, Paul Coupland, Thomas D Otto, Simon R Harris, Thomas R Connor, Anna Bertoni, Harold P Swerdlow, and Yong Gu. 2012. A Tale of Three next Generation Sequencing Platforms: Comparison of Ion Torrent, Pacific Biosciences and Illumina MiSeq Sequencers. *BMC Genomics* 13, 1 (2012), 341. <https://doi.org/10.1186/1471-2164-13-341>
- [221] Uku Raudvere, Liis Kolberg, Ivan Kuzmin, Tambet Arak, Priit Adler, Hedi Peterson, and Jaak Vilo. 2019. G:Profiler: A Web Server for Functional Enrichment Analysis and Conversions of Gene Lists (2019 Update). *Nucleic Acids Research* 47, W1 (July 2019), W191–W198. <https://doi.org/10.1093/nar/gkz369>
- [222] Jenny C.A. Read. 2021. Binocular Vision and Stereopsis Across the Animal Kingdom. *Annual Review of Vision Science* 7, 1 (September 2021), 389–415. <https://doi.org/10.1146/annurev-vision-093019-113212>

- [223] Guido Reina, Hank Childs, Krešimir Matković, Katja Bühler, Manuela Waldner, David Pugmire, Barbora Kozlíková, Timo Ropinski, Patric Ljung, Takayuki Itoh, Eduard Gröller, and Michael Krone. 2020. The Moving Target of Visualization Software for an Increasingly Complex World. *Computers & Graphics* 87, (April 2020), 12–29. <https://doi.org/10.1016/j.cag.2020.01.005>
- [224] Florian Reitz. 2010. A Framework for an Ego-centered and Time-aware Visualization of Relations in Arbitrary Data Repositories. Retrieved February 9, 2024 from <http://arxiv.org/abs/1009.5183>
- [225] Alexander Rind, Wolfgang Aigner, Silvia Miksch, Sylvia Wiltner, Margit Pohl, Thomas Turic, and Felix Drexler. 2011. Visual Exploration of Time-Oriented Patient Data for Chronic Diseases: Design Study and Evaluation. *Information Quality in E-Health* 7058, 301–320. https://doi.org/10.1007/978-3-642-25364-5_22
- [226] Fabio Rizzo. 2022. Optical Immunoassays Methods in Protein Analysis: An Overview. *Chemosensors* 10, 8 (August 2022), 326. <https://doi.org/10.3390/chemosensors10080326>
- [227] James T Robinson, Helga Thorvaldsdóttir, Wendy Winckler, Mitchell Guttman, Eric S Lander, Gad Getz, and Jill P Mesirov. 2011. Integrative Genomics Viewer. *Nature Biotechnology* 29, 1 (January 2011), 24–26. <https://doi.org/10.1038/nbt.1754>
- [228] Hendrik Rohn, Astrid Junker, Anja Hartmann, Eva Grafahrend-Belau, Hendrik Treutler, Matthias Klapperstück, Tobias Czauderna, Christian Klukas, and Falk Schreiber. 2012. VANTED v2: A Framework for Systems Biology Applications. *BMC systems biology* 6, 1 (2012), 1–13. <https://doi.org/10.1186/1752-0509-6-139>
- [229] Armin Ronacher. 2010. Flask — A Micro Web Framework Written in Python. Retrieved May 7, 2024 from <https://flask.palletsprojects.com/>
- [230] Peter J. Rousseeuw. 1987. Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. *Journal of Computational and Applied Mathematics* 20, (November 1987), 53–65. [https://doi.org/10.1016/0377-0427\(87\)90125-7](https://doi.org/10.1016/0377-0427(87)90125-7)

- [231] George P. Rédei. 2008. One Gene-One Enzyme Theorem. *Encyclopedia of Genetics, Genomics, Proteomics and Informatics*, 1396. https://doi.org/10.1007/978-1-4020-6754-9_11856
- [232] George P. Rédei. 2008. Central Dogma. *Encyclopedia of Genetics, Genomics, Proteomics and Informatics*, 309. https://doi.org/10.1007/978-1-4020-6754-9_2672
- [233] George P. Rédei. 2008. Omics. *Encyclopedia of Genetics, Genomics, Proteomics and Informatics*, 1394. https://doi.org/10.1007/978-1-4020-6754-9_11831
- [234] George P. Rédei. 2008. Transcriptome. *Encyclopedia of Genetics, Genomics, Proteomics and Informatics*, 1999. https://doi.org/10.1007/978-1-4020-6754-9_17231
- [235] George P. Rédei. 2008. Proteomics. *Encyclopedia of Genetics, Genomics, Proteomics and Informatics*, 1587–1588. https://doi.org/10.1007/978-1-4020-6754-9_13667
- [236] F. Sanger, S. Nicklen, and A. R. Coulson. 1977. DNA Sequencing with Chain-Terminating Inhibitors. *Proceedings of the National Academy of Sciences* 74, 12 (December 1977), 5463–5467. <https://doi.org/10.1073/pnas.74.12.5463>
- [237] Negar Sarhangi, George P. Patrinos, Hamid Reza Aghaei Meybodi, Shekoufeh Nikfar, Behnaz Sarrami, Pantea Kiani, Bagher Larijani, Nazli Khodayari, and Mandana Hasanzad. 2024. Personalized Medicine and Therapeutics. *A Glimpse at Medicine in the Future*, 111–127. https://doi.org/10.1007/978-981-97-4974-4_7
- [238] A. Savitzky and M. J. E. Golay. 1964. Smoothing and Differentiation of Data by Simplified Least Squares Procedures. 36, 8 (1964), 1627–1639. <https://doi.org/10.1021/ac60214a047>
- [239] Jan Scheer, Alisa Volkert, Nicolas Brich, Lina Weinert, Nandhini Santhanam, Michael Krone, Thomas Ganslandt, Martin Boeker, and Till Nagel. 2022. Visualization Techniques of Time-Oriented Data for the Comparison of Single Patients With Multiple Patients or Cohorts: Scoping Review. *Journal of Medical Internet Research* 24, 10 (October 2022), e38041. <https://doi.org/10.2196/38041>

- [240] M H Schiffman, H M Bauer, A T Lorincz, M M Manos, J C Byrne, A G Glass, D M Cadell, and P M Howley. 1991. Comparison of Southern Blot Hybridization and Polymerase Chain Reaction Methods for the Detection of Human Papillomavirus DNA. *Journal of Clinical Microbiology* 29, 3 (March 1991), 573–577. <https://doi.org/10.1128/jcm.29.3.573-577.1991>
- [241] Christoph Schulz, Arlind Nocaj, Mennatallah El-Assady, Steffen Frey, Marcel Hlawatsch, Michael Hund, Grzegorz Karch, Rudolf Netzel, Christin Schätzle, Miriam Butt, Daniel A. Keim, Thomas Ertl, Ulrik Brandes, and Daniel Weiskopf. 2016. Generative Data Models for Validation and Evaluation of Visualization Techniques. In *Proceedings of the Sixth Workshop on beyond Time and Errors on Novel Evaluation Methods for Visualization (BELIV '16)*, 2016. Association for Computing Machinery, New York, NY, USA, 112–124. <https://doi.org/10.1145/2993901.2993907>
- [242] Christoph Schulz. 2021. Uncertainty-Aware Visualization Techniques. Doctoral dissertation. <https://doi.org/10.18419/OPUS-12115>
- [243] Rainer Schwacke, Gabriel Y. Ponce-Soto, Kirsten Krause, Anthony M. Bolger, Borjana Arsova, Asis Hallab, Kristina Gruden, Mark Stitt, Marie E. Bolger, and Björn Usadel. 2019. MapMan4: A Refined Protein Classification and Annotation Framework Applicable to Multi-Omics Data Analysis. *Molecular Plant* 12, 6 (June 2019), 879–892. <https://doi.org/10.1016/j.molp.2019.01.003>
- [244] Marco Schäfer, Nicolas Brich, Jan Byška, Sérgio M. Marques, David Bednář, Philipp Thiel, Barbora Kozlíková, and Michael Krone. 2023. InVADO: Interactive Visual Analysis of Molecular Docking Data. *IEEE Transactions on Visualization and Computer Graphics* (2023), 1–14. <https://doi.org/10.1109/TVCG.2023.3337642>
- [245] Christin Seifert, Vedran Sabol, and Wolfgang Kienreich. 2010. Stress Maps: Analysing Local Phenomena in Dimensionality Reduction Based Visualisations. In *EuroVAST 2010: International*

- Symposium on Visual Analytics Science and Technology*, 2010. The Eurographics Association. <https://doi.org/10.2312/PE/EuroVAST/EuroVAST10/013-018>
- [246] A.L. Shaffer, W. Wojnar, and W. Nelson. 1990. Amplification, Detection, and Automated Sequencing of Gibbon Interleukin-2 mRNA by *Thermus Aquaticus* DNA Polymerase Reverse Transcription and Polymerase Chain Reaction. *Analytical Biochemistry* 190, 2 (November 1990), 292–296. [https://doi.org/10.1016/0003-2697\(90\)90196-G](https://doi.org/10.1016/0003-2697(90)90196-G)
- [247] Neethu Shah, Haixu Tang, Thomas G. Doak, and Yuzhen Ye. 2010. Comparing Bacterial Communities Inferred from 16S rRNA Gene Sequencing and Shotgun Metagenomics. In *Biocomputing 2011*. World Scientific, 165–176. https://doi.org/10.1142/9789814335058_0018
- [248] Paul Shannon, Andrew Markiel, Owen Ozier, Nitin S. Baliga, Jonathan T. Wang, Daniel Ramage, Nada Amin, Benno Schwikowski, and Trey Ideker. 2003. Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks. *Genome Research* 13, 11 (November 2003), 2498–2504. <https://doi.org/10/b7kgpg>
- [249] Lei Shi, Chen Wang, Zhen Wen, Huamin Qu, Chuang Lin, and Qi Liao. 2015. 1.5D Egocentric Dynamic Network Visualization. *IEEE Transactions on Visualization and Computer Graphics* 21, 5 (May 2015), 624–637. <https://doi.org/10.1109/TVCG.2014.2383380>
- [250] Ben Shneiderman and Aleks Aris. 2006. Network Visualization by Semantic Substrates. *IEEE Transactions on Visualization and Computer Graphics* 12, 5 (September 2006), 733–740. <https://doi.org/10.1109/TVCG.2006.166>
- [251] Ben Shneiderman, Catherine Plaisant, Maxine Cohen, Steven M. Jacobs, and Niklas Elmqvist. 2018. *Designing the User Interface: Strategies for Effective Human-Computer Interaction* (Sixth edition, global edition ed.). Pearson Education Limited, Harlow, Essex, England.

- [252] Ben Shneiderman. 1996. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *Proceedings 1996 IEEE Symposium on Visual Languages*, 1996. IEEE Comput. Soc. Press, Boulder, CO, USA, 336–343. <https://doi.org/10.1109/VL.1996.545307>
- [253] Michael Siegrist. 1996. The Use or Misuse of Three-Dimensional Graphs to Represent Lower-Dimensional Data. *Behaviour & Information Technology* 15, 2 (January 1996), 96–100. <https://doi.org/10.1080/014492996120300>
- [254] Harri Siirtola. 2019. The Cost of Pie Charts. In *2019 23rd International Conference Information Visualisation (IV)*, July 2019. IEEE, Paris, France, 151–156. <https://doi.org/10.1109/IV.2019.00034>
- [255] David Sims, Ian Sudbery, Nicholas E. Illott, Andreas Heger, and Chris P. Ponting. 2014. Sequencing Depth and Coverage: Key Considerations in Genomic Analyses. *Nature Reviews Genetics* 15, 2 (February 2014), 121–132. <https://doi.org/10.1038/nrg3642>
- [256] John Snow. 1855. *On the Mode of Communication of Cholera*. London: John Churchill.
- [257] E.M. Southern. 1975. Detection of Specific Sequences among DNA Fragments Separated by Gel Electrophoresis. *Journal of Molecular Biology* 98, 3 (November 1975), 503–517. [https://doi.org/10.1016/S0022-2836\(75\)80083-0](https://doi.org/10.1016/S0022-2836(75)80083-0)
- [258] Charles Spence, Marianna Obrist, Carlos Velasco, and Nimesha Ranasinghe. 2017. Digitizing the Chemical Senses: Possibilities & Pitfalls. *International Journal of Human-Computer Studies* 107, (November 2017), 62–74. <https://doi.org/10.1016/j.ijhcs.2017.06.003>
- [259] Victor Spirin and Leonid A. Mirny. 2003. Protein Complexes and Functional Modules in Molecular Networks. *Proceedings of the National Academy of Sciences of the United States of America* 100, 21 (October 2003), 12123–12128. <https://doi.org/10.1073/pnas.2032324100>

- [260] Marinus Spreen. 1999. Sampling Personal Network Structures: Statistical Inference in Ego-Graphs. Doctoral dissertation.
- [261] S. S. Stevens. 1970. Neural Events and the Psychophysical Law: Power Functions like Those That Govern Subjective Magnitude Show Themselves in Neurelectric Effects. *Science* 170, 3962 (December 1970), 1043–1050. <https://doi.org/10.1126/science.170.3962.1043>
- [262] Charles D. Stolper, Minsuk Kahng, Zhiyuan Lin, Florian Foerster, Aakash Goel, John Stasko, and Duen Horng Chau. 2014. GLO-STIX: Graph-Level Operations for Specifying Techniques and Interactive eXploration. *IEEE Transactions on Visualization and Computer Graphics* 20, 12 (December 2014), 2320–2328. <https://doi.org/10.1109/TVCG.2014.2346444>
- [263] Jana Strahler. 2020. RNA. *Encyclopedia of Behavioral Medicine*, 1929. https://doi.org/10.1007/978-3-030-39903-0_53
- [264] Ying Sun and Marc G. Genton. 2011. Functional Boxplots. 20, 2 (2011), 316–334. <https://doi.org/10.1198/jcgs.2011.09224>
- [265] Damian Szklarczyk, Andrea Franceschini, Stefan Wyder, Kristofer Forslund, Davide Heller, Jaime Huerta-Cepas, Milan Simonovic, Alexander Roth, Alberto Santos, Kalliopi P. Tsafou, Michael Kuhn, Peer Bork, Lars J. Jensen, and Christian von Mering. 2015. STRING V10: Protein-Protein Interaction Networks, Integrated over the Tree of Life. *Nucleic Acids Research* 43, Database issue (January 2015), D447–452. <https://doi.org/10.1093/nar/gku1003>
- [266] Paolo Di Tommaso, Maria Chatzou, Evan W Floden, Pablo Prieto Barja, Emilio Palumbo, and Cedric Notredame. 2017. Nextflow Enables Reproducible Computational Workflows. *Nature Biotechnology* 35, 4 (April 2017), 316–319. <https://doi.org/10.1038/nbt.3820>
- [267] Warren S Torgerson. 1952. Multidimensional Scaling: I. Theory and Method. *Psychometrika* 17, 4 (1952), 401–419. <https://doi.org/10.1007/BF02288916>

- [268] Cole Trapnell, Adam Roberts, Loyal Goff, Geo Pertea, Daehwan Kim, David R Kelley, Harold Pimentel, Steven L Salzberg, John L Rinn, and Lior Pachter. 2012. Differential Gene and Transcript Expression Analysis of RNA-seq Experiments with TopHat and Cufflinks. *Nature Protocols* 7, 3 (March 2012), 562–578. <https://doi.org/10.1038/nprot.2012.016>
- [269] Charles Truong, Laurent Oudre, and Nicolas Vayatis. 2020. Selective Review of Offline Change Point Detection Methods. *Signal Processing* 167, (2020). <https://doi.org/10.1016/j.sigpro.2019.107299>
- [270] Edward R. Tufte. 1997. *Visual Explanations: Images and Quantities, Evidence and Narrative*. Graphics Press, Cheshire, Conn.
- [271] Edward R. Tufte (Ed.). 2013. *Envisioning Information* (14. print ed.). Graphics Press, Cheshire, Conn.
- [272] Edward R. Tufte. 2018. *The Visual Display of Quantitative Information* (Second edition, tenth printing, April 2018 ed.). Graphics Press, Cheshire, Conn.
- [273] John W Tukey and Paul A Tukey. 1985. Computer Graphics and Exploratory Data Analysis: An Introduction. In *Proceedings of the Sixth Annual Conference and Exposition: Computer Graphics'85*, 1985. National Computer Graphics Association, 773–785.
- [274] John W. Tukey. 1977. *Exploratory Data Analysis* (Repr. ed.). Addison-Wesley, Reading, Mass.
- [275] John W. Tukey. 1980. We Need Both Exploratory and Confirmatory. *The American Statistician* 34, 1 (February 1980), 23–25. <https://doi.org/10.1080/00031305.1980.10482706>
- [276] Mathias Uhlén, Linn Fagerberg, Björn M. Hallström, Cecilia Lindskog, Per Oksvold, Adil Mardinoglu, Åsa Sivertsson, Caroline Kampf, Evelina Sjöstedt, Anna Asplund, IngMarie Olsson, Karolina Edlund, Emma Lundberg, Sanjay Navani, Cristina Al-Khalili Szigyarto, Jacob Odeberg, Dijana Djureinovic, Jenny Ottosson Takanen, Sophia Hober, Tove Alm, Per-Henrik Edqvist, Holger Berling, Hanna Tegel, Jan Mulder, Johan Rockberg, Peter

- Nilsson, Jochen M. Schwenk, Marica Hamsten, Kalle Von Feilitzen, Mattias Forsberg, Lukas Persson, Fredric Johansson, Martin Zwahlen, Gunnar Von Heijne, Jens Nielsen, and Fredrik Pontén. 2015. Tissue-Based Map of the Human Proteome. *Science* 347, 6220 (January 2015), 1260419. <https://doi.org/10.1126/science.1260419>
- [277] Imre Vastrik, Peter D'Eustachio, Esther Schmidt, Geeta Joshi-Tope, Gopal Gopinath, David Croft, Bernard de Bono, Marc Gillespie, Bijay Jassal, Suzanna Lewis, Lisa Matthews, Guanming Wu, Ewan Birney, and Lincoln Stein. 2007. Reactome: A Knowledge Base of Biologic Pathways and Processes. *Genome Biology* 8, 3 (2007), R39. <https://doi.org/10.1186/gb-2007-8-3-r39>
- [278] Corinna Vehlow, Fabian Beck, and Daniel Weiskopf. 2017. Visualizing Group Structures in Graphs: A Survey. *Computer Graphics Forum* 36, 6 (September 2017), 201–225. <https://doi.org/10.1111/cgf.12872>
- [279] Corinna Vehlow, Thomas Reinhardt, and Daniel Weiskopf. 2013. Visualizing Fuzzy Overlapping Communities in Networks. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (December 2013), 2486–2495. <https://doi.org/10.1109/TVCG.2013.232>
- [280] E. F. Vernier, J. L. D. Comba, and A. C. Telea. 2021. Guided Stable Dynamic Projections. 40, 3 (2021), 87–98. <https://doi.org/10.1111/cgf.14291>
- [281] E. F. Vernier, R. Garcia, I. P. da Silva, João Luiz Dihl Comba, and Alexandru C Telea. 2020. Quantitative Evaluation of Time-Dependent Multidimensional Projection Techniques. 39, 3 (2020), 241–252. <https://doi.org/10.1111/cgf.13977>
- [282] George Vernikos, Duccio Medini, David R Riley, and Hervé Tettelin. 2015. Ten Years of Pan-Genome Analyses. *Current Opinion in Microbiology* 23, (February 2015), 148–154. <https://doi.org/10.1016/j.mib.2014.11.016>
- [283] Jasper Verwilt, Pieter Mestdag, and Jo Vandesompele. 2023. Artifacts and Biases of the Reverse Transcription Reaction in RNA

- Sequencing. *RNA* 29, 7 (July 2023), 889–897. <https://doi.org/10.1261/rna.079623.123>
- [284] Markus C. Wahl, Cindy L. Will, and Reinhard Lührmann. 2009. The Spliceosome: Design Principles of a Dynamic RNP Machine. *Cell* 136, 4 (February 2009), 701–718. <https://doi.org/10.1016/j.cell.2009.02.009>
- [285] Q. Wang and R.S. Laramée. 2022. EHR STAR: The State-Of-the-Art in Interactive EHR Visualization. *Computer Graphics Forum* 41, 1 (February 2022), 69–105. <https://doi.org/10.1111/cgf.14424>
- [286] Yong Wang, Qiaomu Shen, Daniel Archambault, Zhiguang Zhou, Min Zhu, Sixiao Yang, and Huamin Qu. 2016. AmbiguityVis: Visualization of Ambiguity in Graph Layouts. *IEEE Trans. Vis. Comput. Graphics* 22, 1 (2016), 359–368. <https://doi.org/10.1109/TVCG.2015.2467691>
- [287] Zhong Wang, Mark Gerstein, and Michael Snyder. 2009. RNA-Seq: A Revolutionary Tool for Transcriptomics. *Nature Reviews Genetics* 10, 1 (January 2009), 57–63. <https://doi.org/10.1038/nrg2484>
- [288] Ziyi Wang, Heng Qiu, Jianbo He, Langxia Liu, Wei Xue, Archa Fox, Jennifer Tickner, and Jiake Xu. 2020. The Emerging Roles of hnRNPK. *Journal of Cellular Physiology* 235, 3 (March 2020), 1995–2008. <https://doi.org/10.1002/jcp.29186>
- [289] Colin Ware. 2008. *Visual Thinking for Design*. Morgan Kaufmann Elsevier Science & Technology, Burlington, Mass. Amsterdam.
- [290] Colin Ware. 2013. *Information Visualization: Perception for Design* (3d edition ed.). Morgan Kaufmann, Waltham, MA.
- [291] Duncan J. Watts and Steven H. Strogatz. 1998. Collective Dynamics of ‘Small-World’ Networks. *Nature* 393, 6684 (June 1998), 440–442. <https://doi.org/10.1038/30918>
- [292] J Welch, P Ford, R Teplick, and R Rubsamen. 1991. The Massachusetts General Hospital-Marquette Foundation Hemodynamic

- and Electrocardiographic Database—Comprehensive Collection of Critical Care Waveforms. *Clinical Monitoring* 7, 1 (1991), 96–97.
- [293] Graham J. Wills. 1999. NicheWorks—Interactive Visualization of Very Large Graphs. *Journal of Computational and Graphical Statistics* 8, 2 (June 1999), 190–212. <https://doi.org/10.1080/10618600.1999.10474810>
- [294] Hsiang-Yun Wu, Martin Nöllenburg, Filipa L Sousa, and Ivan Viola. 2019. Metabopolis: Scalable Network Layout for Biological Pathway Diagrams in Urban Map Style. *BMC bioinformatics* 20, 1 (2019), 1–20. <https://doi.org/10.1186/s12859-019-2779-4>
- [295] Yanhong Wu, Naveen Pitipornvivat, Jian Zhao, Sixiao Yang, Guowei Huang, and Huamin Qu. 2016. egoSlider: Visual Analysis of Egocentric Network Evolution. *IEEE Transactions on Visualization and Computer Graphics* 22, 1 (January 2016), 260–269. <https://doi.org/10.1109/TVCG.2015.2468151>
- [296] Xiao Fan Wang and Guanrong Chen. 2003. Complex Networks: Small-world, Scale-Free and Beyond. *IEEE Circuits and Systems Magazine* 3, 1 (2003), 6–20. <https://doi.org/10.1109/MCAS.2003.1228503>
- [297] Zhao-Ming Xiao, Dao-Jun Lv, Yu-zhong Yu, Chong Wang, Tao Xie, Tao Wang, Xian-Lu Song, and Shan-Chao Zhao. 2021. SMARCC1 Suppresses Tumor Progression by Inhibiting the PI3K/AKT Signaling Pathway in Prostate Cancer. *Frontiers in Cell and Developmental Biology* 9, (June 2021), 678967. <https://doi.org/10.3389/fcell.2021.678967>
- [298] Ke Xu, Ivona Bezakova, Leonid Bunimovich, and Soojin V. Yi. 2011. Path Lengths in Protein–Protein Interaction Networks and Biological Complexity. *PROTEOMICS* 11, 10 (May 2011), 1857–1867. <https://doi.org/10.1002/pmic.201000684>
- [299] Yuewei Xu, Anke Nijhuis, and Hector C. Keun. 2022. RNA-binding Motif Protein 39 (RBM39): An Emerging Cancer Target. *British Journal of Pharmacology* 179, 12 (June 2022), 2795–2812. <https://doi.org/10.1111/bph.15331>

- [300] Mingliang Xue, Yunhai Wang, Chang Han, Jian Zhang, Zheng Wang, Kaiyi Zhang, Christophe Hurter, Jian Zhao, and Oliver Deussen. 2023. Target Netgrams: An Annulus-Constrained Stress Model for Radial Graph Visualization. *IEEE Transactions on Visualization and Computer Graphics* 29, 10 (October 2023), 4256–4268. <https://doi.org/10.1109/TVCG.2022.3187425>
- [301] Rendong Yang, Yun Bai, Zhaohui Qin, and Tianwei Yu. 2014. EgoNet: Identification of Human Disease Ego-Network Modules. *BMC Genomics* 15, 1 (December 2014), 314. <https://doi.org/10.1186/1471-2164-15-314>
- [302] Xinsong Yang, Lei Shi, Madelaine Daianu, Hanghang Tong, Qingsong Liu, and Paul Thompson. 2017. Blockwise Human Brain Network Visual Comparison Using NodeTrix Representation. *IEEE transactions on visualization and computer graphics* 23, 1 (January 2017), 181–190. <https://doi.org/10.1109/TVCG.2016.2598472>
- [303] Jun Ye, Yilin Pang, Xunjun Yang, Chuan Zhang, Lei Shi, Zhitao Chen, Guijia Huang, Xianhe Wang, and Fangyang Lu. 2023. PPIH Gene Regulation System and Its Prognostic Significance in Hepatocellular Carcinoma: A Comprehensive Analysis. *Aging* (October 2023). <https://doi.org/10.18632/aging.205134>
- [304] Zhiyuan Zhang, Arunesh Mittal, Supriya Garg, Alexander Dimitriyadi, I Ramakrishnan, Rong Zhao, Asa Viceddio, and Klaus Mueller. 2010. A Visual Analytics Framework for Emergency Room Clinical Encounters. In *IEEE Workshop on Visual Analytics in Health Care*, 2010.
- [305] Jian Zhao, Michael Glueck, Fanny Chevalier, Yanhong Wu, and Azam Khan. 2016. Egocentric Analysis of Dynamic Networks with EgoLines. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, May 2016. ACM, San Jose California USA, 5003–5014. <https://doi.org/10.1145/2858036.2858488>
- [306] Guangyan Zhou and Jianguo Xia. 2018. OmicsNet: A Web-Based Tool for Creation and Visual Analysis of Biological Networks

in 3D Space. *Nucleic acids research* 46, W1 (2018), W514–W522. <https://doi.org/10.1093/nar/gky510>

- [307] Hong Zhou, Panpan Xu, Xiaoru Yuan, and Huamin Qu. 2013. Edge Bundling in Information Visualization. *Tsinghua Science and Technology* 18, 2 (April 2013), 145–156. <https://doi.org/10.1109/TST.2013.6509098>