BENEDIKT GÜNTHER BRINK

OMICS VISUALIZATION AND ITS APPLICATION TO PRESYMPTOMATIC DIAGNOSIS OF ORAL CANCER

OMICS VISUALIZATION AND ITS APPLICATION TO PRESYMPTOMATIC DIAGNOSIS OF ORAL CANCER

BENEDIKT GÜNTHER BRINK

Dissertation
Faculty of Technology, Bielefeld University
In partial fulfillment of the requirements for the degree of
Doctor rerum naturalium (Dr. rer. nat.) in the subject of Bioinformatics

Supervised by
Prof. Dr.-Ing. Tim W. Nattkemper
Prof. Dr. Ryan R. Brinkman
Dr. Stefan P. Albaum

Bielefeld, December 2017

Benedikt Günther Brink: *Omics visualization and its application to pre-symptomatic diagnosis of oral cancer,* Dissertation, © December 2017

Science, my lad, has been built upon many errors;
but they are errors which it was good to fall into,
for they led to the truth.

— Jules Verne (1864)

ABSTRACT

About 30 zettabytes ($30 \cdot 10^{21}$ bytes) of data are generated worldwide every second — so much that over 90 % of the data in the world today has been created in the last two years alone. Science as well is flooded by an ever increasing amount of data. However, accessing the information hidden in this massive amount of data is a challenging task and in science often presents a hindrance to knowledge discovery. One way to overcome this is a good visualization, which can greatly support people and scientists in exploring, understanding, and enjoying data. In this thesis, I present three examples for a task oriented visualization in some of the most data-rich disciplines in science: biochemistry, healthcare, and biology.

The first example is situated in the field of biochemistry. Since the 1980s, natural sciences challenged educational institutions and media to keep the society on an appropriate level of knowledge and understanding. By investigating the potential of infographics, graphical design, and game motivation, I present a mnemonic card game based on creative design to aid the learning of a special group of biomolecules, the amino acids. Each amino acid is composed of a number of features. The latter are intuitively encoded into shapes, colors, and textures to assist our abilities in interpreting visual stimuli. Thus, it facilitates recognizing such features, grouping them, noting relationships, and ultimately memorizing the structural formulas. The cards translate complex molecular structures into visual formats that are both easier to assess and to understand. The result is a unique teaching tool that is not only subject-oriented, fun, and engaging, but also helps students retain relevant information such as properties and formulas through perceptual memory.

The second example tackles a problem from the field of healthcare. Oral cancer has a major impact worldwide, accounting for 274 000 new cases and 145 000 deaths each year, making it the sixth most common cancer. Developing methods for the detection of cancer in its earliest stages can greatly increase the chances for a successful treatment. Many cancers (including oral cancer) are known to development through multiple steps, which are caused by certain mutations to the genome. A recently published protocol by Hughesman et al. (2016) describes means for high-throughput detection of these mutations using droplet digital PCR. However, methods for automated analysis and visualization of this data are unavailable. In this thesis, I present *ddPCRclust*, an R package for automated analysis of droplet digital PCR data. It can automatically analyze and visualize data from droplet digital PCR experiments with up to four targets per

reaction in a non-orthogonal layout. Results are on a par with manual analysis, but only take minutes to compute instead of hours. The accompanying Shiny application *ddPCRvis* provides easy access to the functionalities of *ddPCRclust* through a web-browser based graphical user interface, enabling the user to interactively filter data and change parameters, as well as view and modify results.

The third example involves some of the most data-rich disciplines in biology - transcriptomics, proteomics, and metabolomics. *Omics Fusion* is a web based platform for the integrative analysis of omics data. It provides a collection of new and established tools and visualization methods to support researchers in exploring omics data, validating results, or understanding how to adjust experiments in order to make new discoveries. It is easily extendible and new visualization methods are added continuously. I present an example for a task-oriented visualization of functional annotated omics data based on the established Clusters of Orthologous Groups (COG) database and gene ontology (GO) terms.

ZUSAMMENFASSUNG

Rund 30 Zettabyte ($30 \cdot 10^{21}$ Byte) an Daten werden tagtäglich in der Welt generiert — so viel, dass mehr als 90 % der Daten heutzutage allein in den letzten zwei Jahren erzeugt wurden. Dieser Trend macht auch vor den Naturwissenschaften nicht halt. Die Informationen zu extrahieren, die in dieser Datenflut versteckt sind, stellt eine große Herausforderung dar und verlangsamt häufig die Forschungsarbeit. Eine Möglichkeit Abhilfe zu schaffen ist eine gute Visualisierung, welche Menschen und Wissenschaftler darin unterstützt, ihre Daten zu sondieren, zu verstehen und zu analysieren. In dieser Dissertation präsentiere ich drei Beispiele für Visualisierungen in besonders datenreichen Wissenschaften: Biochemie, Medizin und Biologie.

Das erste Beispiel ist im Bereich der Biochemie angesiedelt. Seit den 1980er Jahren stellen die Fortschritte in den Naturwissenschaften sowohl die Bildungseinrichtungen, als auch die Gesellschaft im Allgemeinen vor die Herausforderung, mit ihnen Schritt zu halten. Ich untersuche das Potential von Infografiken, grafischem Design und Spielmotivation anhand eines mnemonischen Kartenspiels über Aminosäuren. Das Kartenspiel basiert auf dem Prinzip eines klassichen Quartett-Spiels und soll das lernen der Aminosäuren und einiger ihrer grundlegenden biochemischen Eigenschaften unterstützen. Jede Karte besteht aus einer Reihe von Merkmalen, welche in Formen, Farben und Strukturen kodiert sind, um die angeborenen Fähigkeiten der Menschen im Bezug auf visuelle Stimuli zu nutzen. Die Karten übersetzen komplexe biochemische Sturkturformeln in ein visuelles Format, was einfacher zu erkennen und zu verstehen ist. Das Resultat

ist ein einzigartiges Lerninsturment, was eine anwendungsorientierte Visualisierung mit dem Spaß einen Kartenspiels verbindet und so das Lernen der Informationen vereinfacht.

Das zweite Beispiel befasst sich mit einem Problem aus dem Bereich der Medizin. Mundhöhlenkarzinoma sind die sechsthäufigste Form von Kebs auf der Welt, verantwortlich für 274 000 Neuerkrankungen und 145 000 Tode jährlich. Methoden für die Früherkennung von Krebs können die Chancen für eine erfolgreiche Bahndlung signifikant erhöhen. Es ist bekannt, dass sich viele Krebsarten (einschließlich Mundhöhlenkarzinoma) durch einen mehrstufigen Prozess entwickeln, welche durch bestimmte Mutationen im Genom ausgelöst werden. Das Manuskript von Hughesman u. a. (2016) beschreibt ein Verfahren zum Screening nach diesen Mutationen mit Hilfe von droplet digital PCR. Automatische Verfahren zur Analyse von diesen Daten sind jedoch nicht verfügbar. Ich präsentiere *ddPCRclust*, ein R Paket für die automatische Analyse droplet digital PCR Daten. Es kann Daten mit bis zu vier Biomarkern pro Reaktion automatisch analysieren und visualisieren. Die Ergebnisse der automatischen Analyse sind vergleichbar mit der manuellen Analyse durch Experten, sie ist jedoch innerhalb von wenigen Minuten abgeschlossen anstatt mehrerer Stunden. Darüber hinaus gibt es eine begleitende Shiny Anwendung *ddPCRvis*, welche eine grafische Benutzeroberfläche für *ddPCRclust* im Webbrowser bereitstellt. Nutzer können so ihre Daten interaktiv filtern, Parameter anpassen und die Resultate sowohl betrachten, als auch modifizieren.

Das dritte Beispiel umfasst einige Bereiche der Biologie, in denen mit die größten Datenmengen generiert werden — Transriptomics, Proteomics, und Metabolomics. *Omics Fusion* ist eine webbasierte Platform für die integrative Analyse von omics-Daten. Es bietet eine Reihe von neuen und etablierten Werkzeugen und Visualisierungen, um Wissenschaftler in ihrer Arbeit zu unterstützen. Das Hauptaugenmerk liegt dabei auf der Datenanalyse, z.B. dem Validieren von Hypothesen oder dem Entdecken von unerwarteten Mustern. Dabei wird *Omics Fusion* fortlaufend durch neue Methoden zur Visualisierung oder Analyse erweitert. Ich stelle ein Beispiel für eine solche anwendungsorientierte Visualisierung in *Omics Fusion* anhand von funktional annotierten omics Daten basierend auf der etablierten Clusters of Orthologous Groups (COG) Datenbank und den gene ontology (GO) terms vor.

PUBLICATIONS

Brink, Benedikt G, Justin Meskas, and Ryan R Brinkman (2018). "ddPCRclust: an R package and Shiny app for automated analysis of multiplexed ddPCR data." In: *Bioinformatics*.

Brink, Benedikt G, Annica Seidel, Nils Kleinbölting, Tim W Nattkemper, and Stefan P Albaum (2016). "Omics Fusion–A Platform for Integrative Analysis of Omics Data." In: *Journal of integrative bioinformatics* 13.4, pp. 43–46. doi: 10.2390/biecoll-jib-2016-296.

Hattab, Georges, Benedikt G Brink, and Tim W Nattkemper (June 2016). "A mnemonic card game for your amino acids." In: *Information+ Conference*. doi: 10.5281/zenodo.55101.

Tsai, Chia-Hong, Krzysztof Zienkiewicz, Cynthia L Amstutz, Benedikt G Brink, Jaruswan Warakanont, Rebecca Roston, and Christoph Benning (2015). "Dynamics of protein and polar lipid recruitment during lipid droplet assembly in Chlamydomonas reinhardtii." In: *The Plant Journal* 83.4, pp. 650–660. doi: 10.1111/tpj.12917.

ACKNOWLEDGMENTS

My time as a student at Bielefeld University started in the fall of 2008 and since that time I met an innumerable number of great people — each and everyone somehow altering my path, which eventually led me to where I am today. I would like to start these acknowledgments with my roommate for the first five-and-a-half years and friend since high school, Sebastian Grenz. I also thank Sebastian Kral for many memorable party nights, Torsten 'neocon' Hübner for being my football buddy both with and without cars, and Natalie Frese for introducing me to Bingo. I thank Daniel Blume, Franziska Obracaj, Edward Bock, and Carina Domzalski for the double sandwich and making my time as a student the amazing experience that it was.

In 2013 I became a PhD candidate in the best graduate school in the world, "Computational Methods for the Analysis of the Diversity and Dynamics of Genomes" (DiDy). I want to thank all my colleagues from that time, especially Georges Hattab for sharing an office with me for three years, Guillaume Holley, Violette Reviere, Nina Luhmann, and Benedikt Löwes for sharing a house in Vancouver for six months, Tina Zekic, Markus Lux, Lukas Pfannschmidt, Liren Huang, Jia Yu, Pina Krell, Omar Castillo, Kostas Tzanakis, Linda Sundermann, and all the others for many discussions, after work beers, and countless hours of ping-pong training. Furthermore, I would like to thank Roland Wittler for coordinating the graduate school perfectly, organizing everything that needs to be organized, reminding everyone who needs to be reminded, and on top of that guiding us as PhD candidates.

In addition, the DiDy graduate school enabled my exchange with Vancouver for six months in 2015, where I met Justin Meskas, who turned out to be not only a proficient colleague, but also a good friend.

I was also very lucky to have fantastic supervisors during my time as a PhD student, namely Tim Nattkemper, Ryan Brinkman, and Stefan Albaum. You welcomed me in your research groups with open arms and provided me the guidance (and at times the warnings) that I needed to achieve this thesis.

Last but not least, I would like to thank my family, especially my mom Gisela Brink, who surely did not have the easiest life, but always managed to be there for her kids, and my grandfather Günther Brink, whose attitude towards life has been a great inspiration for me.

Writing this thesis has been one of the biggest challenges in my life, but the road has been peppered with a lot of fun, amazing adventures, true friends, and great colleagues.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

AI	allelic imbalance
ANOVA	analysis of variance
CNA	copy number aberration
CNV	copy number variation
COG	Clusters of Orthologous Groups
CPDs	copies per droplet
CSS	Cascading Style Sheets
CSV	comma-separated values
DNA	deoxyribonucleic acid
ddPCR	droplet digital PCR
FFPE	formalin-fixed paraffin-embedded
GO	gene ontology
GUI	graphical user interface
HTML	Hypertext Markup Language
IQR	interquartile range
KOG	eukaryotic orthologous groups
LOH	loss of heterozygosity
mRNA	messenger RNA
MVC	model-view-controller
OPL	oral premalignant lesion
ORF	open reading frame

PCA	principial component analysis
PCR	polymerase chain reaction
PEST	Public Engagement of Science and Technology
PUS	Public Understanding of Sciences
RNA	ribonucleic acid
SBML	Systems Biology Markup Language
SCC	squamous cell carcinoma
SNP	single-nucleotide polymorphism
SVG	Scalable Vector Graphic
TCGA	the cancer genome atlas
UV	ultraviolet

Part I

DATA MINING AND VISUALIZATION

The first part of this thesis contains three chapters and introduces the concepts of data, data mining, and visualization. It starts off with a brief history of data and visualization, before going into detail on what types of visualizations exist, why they are useful and how to use them effectively. The second chapter motivates this thesis based on the previously introduced concepts and outlines the structure of this thesis. The third chapter presents an example for a task oriented visualization in form of a card game.

1 INTRODUCTION

In this chapter, I introduce the concept of data and its visual abstraction, motivated by a brief glance into history. I present what comprises data visualization, why data abstraction is necessary, and how to create a good visualization.

1.1 A BRIEF HISTORY OF DATA AND VISUALIZATION

The scientific revolution in Europe between the 16th and 18th century marked the birth of modern science as we know it today. It revolutionized the fundamental process of thinking for many people in the western world, inspiring free thought and what we now call the Age of Enlightenment. In its wake a need for something arose, which was until then often neglected: data. Hypotheses needed to be tested and validated, which required evidence — reproducible observations of the real world, which were no longer based on religious beliefs or traditional practice. The term data in general describes facts and statistics collected together for reference or analysis. Thus, data is a set of qualitative or quantitative variables.

As the name suggests, quantitative data contains information about quantities, i. e. information that can be measured and written down with numbers. An example of quantitative data would be temperature measurements of a weather station. Qualitative data on the other hand is information that can not actually be measured, for example the historic accounts of a contemporary witness.

While the latter is usually easy to understand and interpret by humans, the former might not. Several steps could be necessary for any information hidden within the data to become obvious. *Raw data*, defined as a collection of information before it has been curated or transformed by researchers, needs to be examined and/or corrected. This could involve the removal of outliers, instrument errors, or data entry errors; the translation of the data into a different format, or statistical operations like normalization. The result of these operations is called *processed data* and often helps in various data related tasks (e. g. data comparison). Such a result can be further analyzed using statistics or — with the support of computers — databases, machine learning algorithms, etc. In turn, this processed data is employed to create data abstractions (that are independent of the initial domain knowledge) so as to be visually encoded and visualized, which can highlight different aspects of the data and thereby foster knowledge discovery.

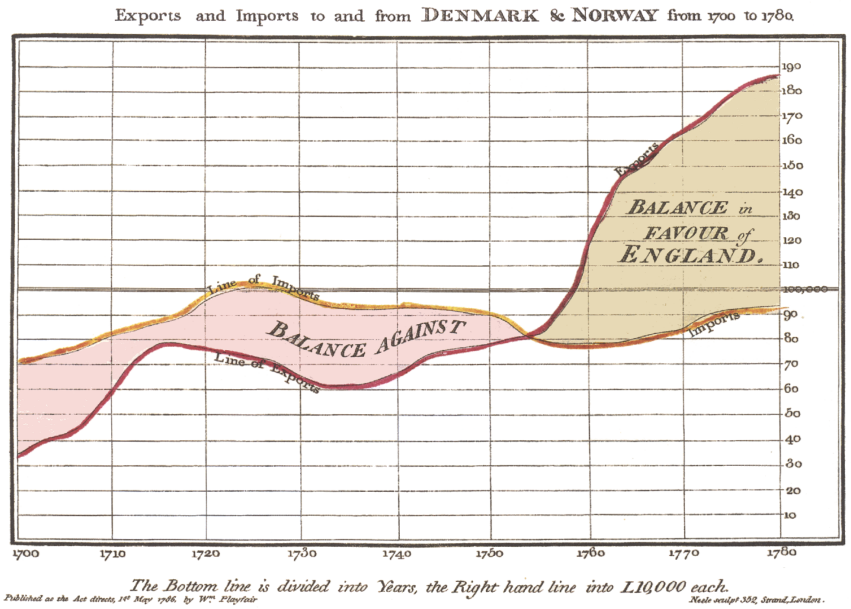


Figure 1.1: England's trade-balance with Denmark and Norway from 1700 to 1780 in a time-series chart, published by William Playfair in 1786.

The whole process from raw data to knowledge discovery is defined as *data mining*.

With data becoming more widespread, data mining became essential. The numbers needed to be understood, interpreted, and shared. One of the early pioneers in this area of research was William Playfair, who is often cited as the founder of graphical methods of statistics. He is attributed with inventing several types of diagrams, ranging from line, area, and bar charts to pie charts and circle graphs to show part-whole relations. An example of his early work is given in Figure 1.1 (Playfair, 1801).

Data visualizations are not limited to economics or statistics. The right visual representation can foster the discovery of hidden patterns in the data. One of the earliest examples for this goes back to 1854. London was hit by the third severe outbreak of cholera in 22 years. The widely accepted theory at the time was that cholera was caused by so called 'bad air', but physician John Snow was skeptical about it. During the Soho epidemic in 1854 he created the famous dot map (Figure 1.2) to illustrate the cluster of cholera cases around a certain water pump. Snow's efforts to connect the incidence of cholera with potential geographic sources was based on creating what is now known as a Voronoi diagram. He mapped the locations of individual water pumps and generated cells, which represented all the points on his map which were closest to each available water source. The section of Snow's map being closest to the Broad Street pump was

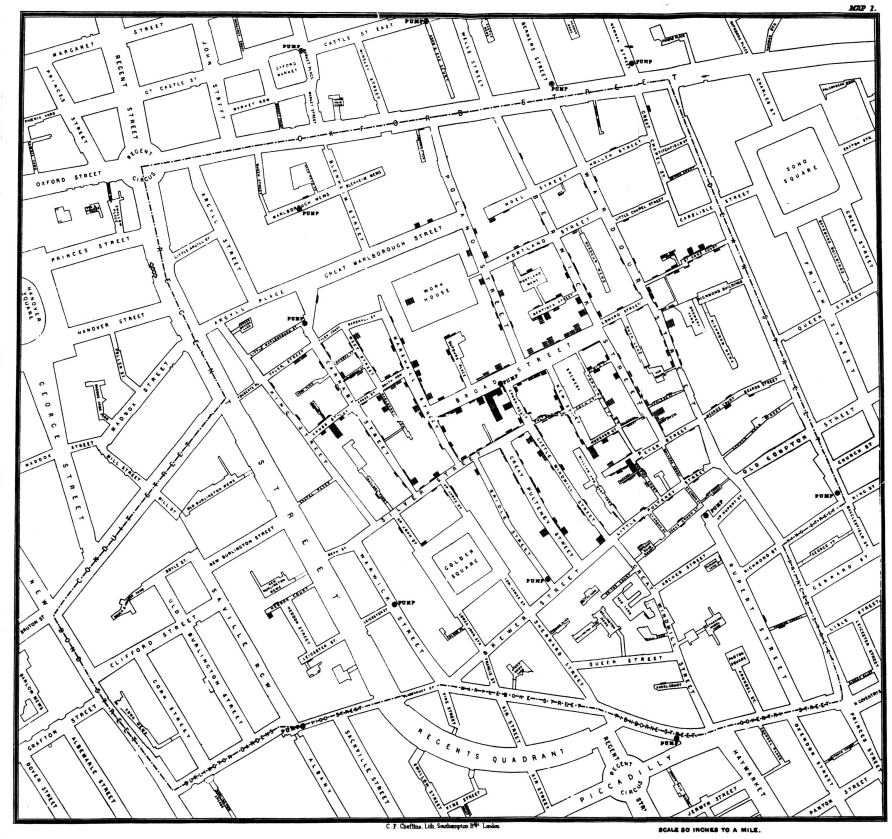


Figure 1.2: Original map by John Snow showing the clusters of cholera cases (indicated by stacked rectangles) in the London epidemic of 1854. The water pump with contaminated water is located at the intersection of Broad Street and Cambridge Street, running into Little Windmill Street.

linked to the highest incidence of cholera cases. Moreover, he used statistics to compare fatalities among the customers of London's different water suppliers, and to illustrate the connection between the quality of the source of water and the number of cholera cases. This convinced the local authorities to disable the pump by removing its handle, effectively ending the outbreak.

We have certainly come a long way, yet the amount of data that surrounds us in today's digital world makes effective data mining and visualization more important than ever. About 30 zettabytes ($30 \cdot 10^{21}$ bytes) of data are generated worldwide every second — so much that over 90 % of the data in the world today has been created in the last two years alone (*The World's Data*). This includes data from all areas, such as weather sensors, social media posts, or Google queries. Data has been described as the new oil of the digital economy (Toonders, 2014).

1.2 DESIGNING A VISUALIZATION

In the context of this thesis, I will focus on mining and visualizing scientific data — specifically biological, biochemical and medical data. Visualization is viewed by many disciplines as a modern equivalent of visual communication. At the start of every visualization project, it is helpful to answer some basic questions:

- What is to be visualized?
- Why visualize it?
- How to visualize it?

1.2.1 *What is to be visualized*

The most important step at the start of a visualization project is to understand what kind of data is available. This defines what information can be extracted from it, and what questions can be answered or what problems can be solved with that information. There are four basic types of datasets: tables, networks, fields and geometry. Additional types might be clusters, sets, or lists. Each dataset type in turn consists of different data types. Certain data types can be categorical or ordered, sequential or cyclic. An overview of the different possible data types is given in Figure 1.3.

In this thesis, we will mostly deal with tables and clusters. Each cell in a table is fully defined by the combination of row and column and contains a value for that pair. For example, in Part II of the thesis I present a specific use case of data mining and visualization in the medical field, i. e. the study and treatment of tumors. The available raw data for this project are tables, where each row represents an item of data and each column an categorical attribute of the dataset. I process the data, filter out certain items, and add another attribute specifying the cluster-membership of each item. Clusters are a special data type, where items within one cluster are more similar to each other than to ones in another cluster. The processed data can be further analyzed and visualized according to the questions at hand. Using these terms, it is possible to abstract data and discuss it, without knowing specifically what a dataset is about.

However, semantics certainly matter. Knowing what each data item represents dictates what kind of questions can be answered later. In Table 1.1 the well-known *mtcars* example table from the R programming language is shown. It might be possible to understand that this table contains data about cars, but without additional information, it is hard to grasp what these values mean and what kind of actions can be performed on such a dataset. A more detailed overview over these actions will be presented in Section 1.2.2.

Figure 1.3: What can be visualized: data, datasets, and attributes. The four basic dataset types are tables, networks, fields and geometry; other possible collections of items include clusters, sets, and lists. These datasets are made up of different combinations of the five data types: items, attributes, links, positions, and grids. For any of these dataset types, the full dataset could be available immediately in the form of a static file, or it might be dynamic data processed gradually in the form of a stream. The type of an attribute can be categorical or ordered, with a further split into ordinal and quantitative. The ordering direction of attributes can be sequential, diverging, or cyclic. (Reproduced from Munzner, 2014)

Table 1.1: Excerpt from the *mtcars* example table in R

mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
21	6	160	110	3.9	2.62	16.46	0	1	4	4
21	6	160	110	3.9	2.875	17.02	0	1	4	4
22.8	4	108	93	3.85	2.32	18.61	1	1	4	1
21.4	6	258	110	3.08	3.215	19.44	1	0	3	1

1.2.2 *Why visualize it*

In general, visualizations provide visual representations of datasets designed to help users carry out tasks more efficiently. The primary goal of any data visualization is to communicate information clearly and efficiently. To solve a particular analytical task, such as understanding causality or making comparisons, the design principle of the graphic must follow the task. This means explicitly designing the visualization with the task abstraction already in mind. In order to achieve that, it is important to understand the nature of the task at hand. Figure 1.4 gives an overview of the different reasons for using a visualization in terms of actions and targets.

Actions are defined as user goals. The users can use a visualization: to analyze (i. e. consume existing or produce additional data), to search, or to query. The target defines the aspect that is of interest to the users. Targets are nouns, whereas actions are verbs. For instance, in Part II of this thesis I want to present the results of a certain analysis to the users. The target is the whole dataset and the users will consume this presentation. In Part III of the thesis on the other hand, I want to enable the users to explore their data. They need to be able to search, query, and analyze both all data, as well as parts of it. Thus, each visualization needs to have its own actions and targets.

To come back to the example in Table 1.1, the information that is missing in order to create meaningful tasks is presented in Table 1.2. Together with the information that each row is linked to a certain car model (not shown), it becomes possible to define actions, such as discovering which car has the best miles/gallon ratio, filter out all

Table 1.2: Additional information for the *mtcars* example

mpg	Miles/(US) gallon	qsec	1/4 mile time
cyl	Number of cylinders	vs	0 = V engine, 1 = straight
disp	Displacement (cu.in.)	am	0 = automatic, 1 = manual
hp	Gross horsepower	gear	Number of forward gears
drat	Rear axle ratio	carb	Number of carburetors
wt	Weight (1000 lbs)		

cars that have manual transmission, search for a specific model and display its values, etc. For each scenario, different task abstractions need to be defined, in order to create a fitting data abstraction for the best visualization. The means how to do so once the data and task abstractions are defined, are presented in Section 1.2.3.

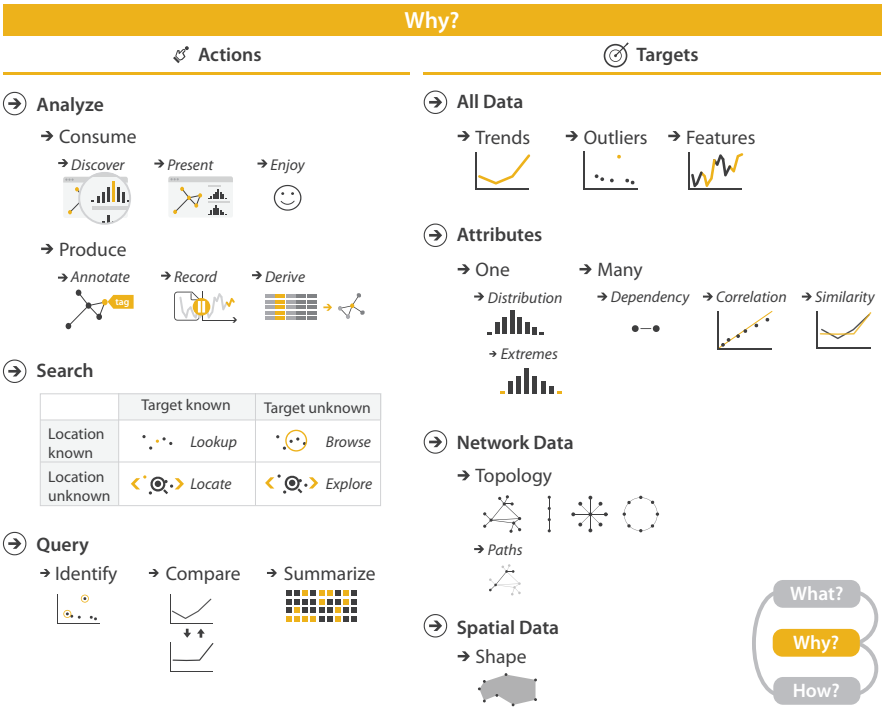


Figure 1.4: Why people are using visualization in terms of actions and targets. The highest-level actions are to use visualizations to consume or produce information. The cases for consuming are to present, to discover, and to enjoy; discovery may involve generating or verifying a hypothesis. At the middle level, search can be classified according to whether the identity and location of the target is known or not: both are known with lookup, the target is known but its location not for locate, the location is known but the target is not for browse, and neither the target nor the location is known for explore. At the low level, queries can have three scopes: identify one target, compare some targets, and summarize all targets. Targets for all kinds of data are finding trends and outliers. For one attribute, the target can be one value, the extremes of minimum and maximum values, or the distribution of all values across the entire attribute. For multiple attributes, the target can be dependencies, correlations, or similarities between them. The target with network data can be topology in general or paths in particular, and with spatial data the target can be shape. (Reproduced from Munzner, 2014)

1.2.3 *How to visualize it*

We can use certain visual encodings of data in order to make it easier for the human brain to acquire the information we want to communicate. A good visualization enables users to process the information much faster. When they are also able to interact with it, as it is possible with a computer based visualization, the users can explore data, gain new insights, and make discoveries that might not have been possible otherwise. I define one distinct approach to create and manipulate a visual representation as a visual *idiom*. The design space of visual idioms is huge. It ranges from static idioms that have a long history, as presented in Section 1.1, to more complicated idioms that interactively link different visual representations using the means of modern, computer based visualizations. For example, moving the cursor over a dot in a scatterplot could highlight the respective element in a different plot, e. g. a bar in a bar chart.

Assigning a property such as color, size, shape, or motion to a data attribute is called mapping. However, visual representation can also be overwhelming and therefore it has to be chosen carefully, which mapping is appropriate for which type of information.

The human eye and the corresponding part in the brain responsible for processing visual information have evolved over millions of years. During that time, certain visual cues have proven to be more important than others. Following the laws of natural selection, whatever quality ensured the survival of an individual was passed on to the next generation (Darwin, 1859). Therefore, movement has to be considered the most prominent aspect of any visual idiom. If one item moves and the rest does not, the human brain immediately notices this, since it could be prey — or a predator. Spotting a predator or prey earlier can be a matter of life and death, which means the pressure of natural selection is especially high in this regard. But humans and their relatives evolved even further, not only detecting shape and movement like most mammals, but also color. Being omnivorous hunter-gatherers, detecting colors is an advantage when looking for fruits and other edible things. This is why birds have an even better color vision than humans, being able to detect four or more dimensions of color, while us humans are limited to three. We can distinguish short wavelengths (blue), middle wavelengths (green), or long wavelength (red), as well as luminance. Neural networks add and subtract the signals when passing the information on to the brain. This causes the formation of three distinct color-opponent-channels: red-green, yellow-blue, and black-white. Ware (2010) details this as the *opponent process theory*, but the original idea can be traced back to Hering, 1878.

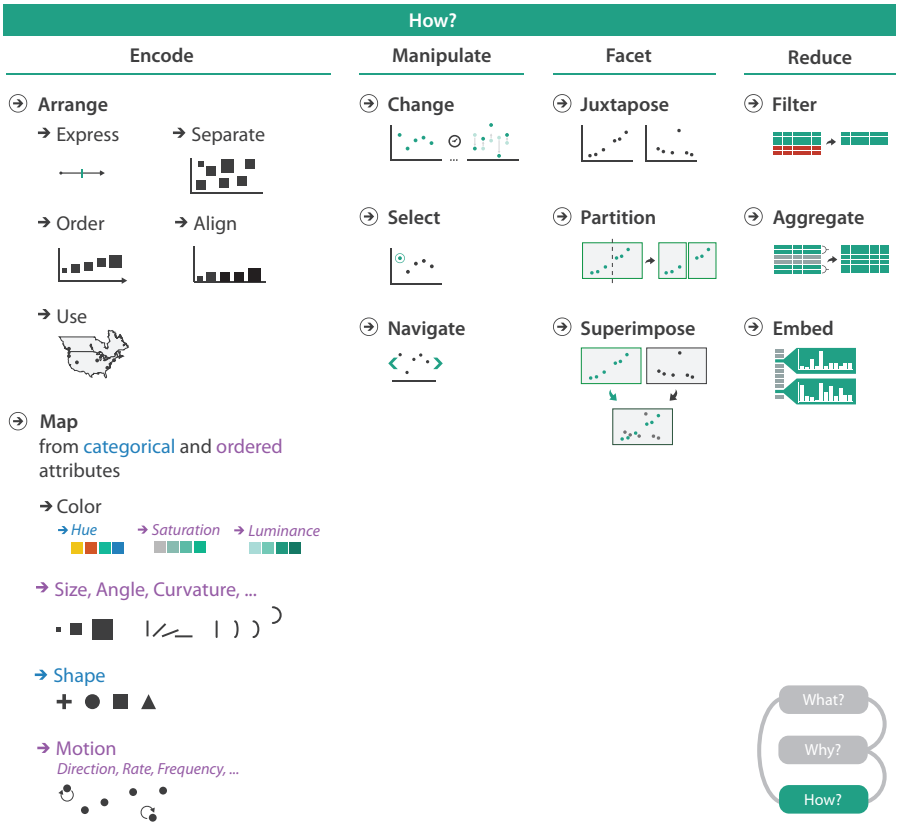


Figure 1.5: How to design visual idioms: encode, manipulate, facet, and reduce. The family of how to encode data within a view has five choices for how to arrange data spatially: express values; separate, order, and align regions; and use given spatial data. This family also includes how to map data with all of the nonspatial visual channels including color, size, angle, shape, and many more. The manipulate family has the choices of change any aspect of the view, select elements from within the view, and navigate to change the viewpoint within the view — an aspect of change with a rich enough set of choices to merit its own category. The family of how to facet data between views has choices for how to juxtapose and coordinate multiple views, how to partition data between views, and how to superimpose layers on top of each other. The family of how to reduce the data shown has the options of filter data away, aggregate many data elements together, and embed focus and context information together within a single view. (Reproduced from Munzner, 2014)

2

THESIS OVERVIEW

In this chapter, I clarify the motivation behind this thesis based on the introduction presented in Chapter 1. Furthermore, I outline the overall structure of the thesis.

2.1 MOTIVATION OF THE THESIS

In Chapter 1 I introduced the concept of visualization and one of its pioneers, William Playfair. He wrote in his book "The Commercial and Political Atlas":

> Men in general are very slow to enter into what is reckoned a new thing; and there seems to be a very universal as well as great reluctance to undergo the drudgery of acquiring information that seems not to be absolutely necessary (Playfair, 1801).

While he did not yet understand or go into detail as to why this is case, his observations are valid. It has been studied intensively that finding and acquiring information from texts, tables, or other uniform sources is difficult and much less effective than using visual cues (Card, Mackinlay, and Shneiderman, 1999; Tufte, 1990; Ware, 2010). The reason for this is that humans are visual beings. Almost half of our brain is dedicated to the visual sense, highly capable of detecting and interpreting any kind of graphical pattern. In fact, our visual brain is so powerful that even the fastest computers using the most advanced algorithms cannot rival a single human brain in terms of image recognition. Millions of years of evolution have yielded visual supercomputers. This becomes evident almost anywhere in nature. Camouflage exists in a wide variety in the animal kingdoms — and the better the camouflage, the better the vision needs to be. Naturally, this does not only include the eyes, but encompasses the corresponding visual cortex in the brain as well, where the visual information needs to processed.

Even though humans today are no longer dependent on hunting or gathering their food, while avoiding other, potentially dangerous predators, these aspects nonetheless shaped the world we live in today. Artificial light makes sure that we can use our primary sense day and night. Crucial information is communicated using shape and color, for instance whether or not it is safe to cross a traffic light. Advertisement makes use of polished images rather than descriptive text, and basic symbols like exit or men's/women's washroom are known

around the world. This is all summarized by the common saying: *A picture is worth a thousand words*.

In a world increasingly flooded with data (see Chapter 1), accessing the visual processing power of the brain is important. Computers can support this by preprocessing data and/or finding predefined patterns. However, knowledge discovery often requires experts in the loop, who can apply their domain knowledge. This is especially true in the field of natural sciences. Creating a powerful visualization, which enables scientists to explore and interpret the data in their own ways, can make all the difference. In the age of computer based visualizations, interactivity is a key aspect in this endeavor (Figure 2.1). It gives the users the power to adjust a visualization towards their need, because in most cases it is simply impossible to display all the information at once. The Visual Information-Seeking Mantra by Shneiderman (1996) summarizes this as follows:

> Overview first, zoom and filter, then details-on-demand.

However, using interactivity is not always useful or possible. In this thesis, I investigate the visual approach and different levels of interactivity in some of most data-rich disciplines, the fields of biology, biochemistry, and healthcare (Raghupathi, 2016).

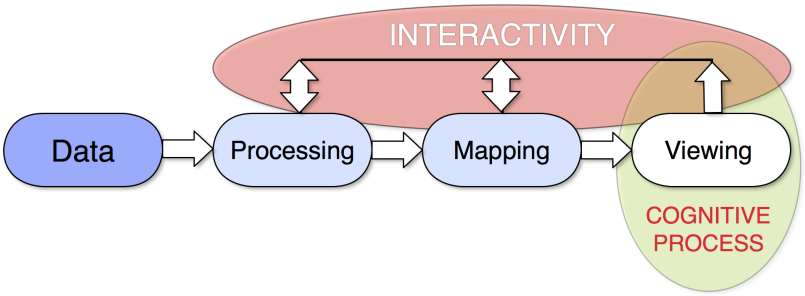


Figure 2.1: Interactivity is an important aspect of data visualization. Data processing (e. g. filtering or clustering), mapping (e. g. color or shape), and the actual visualization should be dynamic.

2.2 STRUCTURE OF THE THESIS

Based on the background presented in Chapter 1 and the motivation highlighted in Chapter 2, Chapter 3 will start with the least interactive visualization — a design for a physical card game. I propose a mnemonic card game based on creative design, which features visual abstractions of amino acids in order to aid in learning and memorizing them. It was presented and exhibited at the Information+ Conference 2016 (Hattab, Brink, and Nattkemper, 2016).

Part II describes a specific medical problem, namely the presymptomatic discovery of oral cancer, which can be solved with the help of

tailored algorithms (Brink, Meskas, and Brinkman, 2018). I present the biological background of this problem in Chapter 4, the algorithmic part of the solution in Chapter 5 and Chapter 6, and the visualization part of the solution in Chapter 7. Combined, it enables scientists and physicians to understand the steps involved in the algorithmic solution and explore the results visually, in turn supporting them in making a diagnosis. As a computer based approach, it already has a higher level of interactivity as a card game, but is limited by the specific biomedical question and by the technology used to answer it. I evaluate the results of the algorithm in Chapter 6 and discuss the benefits and limitations of such a targeted approach Chapter 7. Furthermore, I raise the question whether a more open, exploratory approach, which can be applied on a variety of data without a specific problem at hand, can overcome those.

In Part III, I present Omics Fusion, a new web-based platform for integrative analysis of *omics* data (Brink et al., 2016). Omics is an umbrella term for different fields of biological research, which are introduced in detail in Chapter 8. An approach to visualize functional annotated omics data is presented in Chapter 9. I discuss the benefits and limitations of an open, exploratory approach of analyzing and visualizing biological data and compare it to the previous, targeted approach. Finally, I conclude the thesis in Chapter 10.

3

A MNEMONIC CARD GAME FOR YOUR AMINO ACIDS

Data visualizations do not necessarily need to be about knowledge discovery. They can also be about memorizing existing knowledge, fun, or simply being artistic. In this chapter, I present a static, non interactive visualization of amino acids in the form of a physical card game, in order to facilitate memorizing the amino acids and some of their important physicochemical properties. Part of this chapter have been published under Hattab, Brink, and Nattkemper (2016). The complete set of cards, rules and possible scenarios are available under `https://github.com/ghattab/amino-acids-card-game/`.

3.1 BACKGROUND

Since the 1980s the progression in natural sciences challenged the educational institutions and media to keep the society on an appropriate level of knowledge and understanding. Two very prominent early developments were Public Understanding of Sciences (PUS) based on a report by Bodmer (1986) and Public Engagement of Science and Technology (PEST) by Irwin and Wynne (2003). In 2005, Wynne argued PEST is a more viable solution where public engagement occurs through a dialog among scientists and the public. With efforts from both sides, many areas remain ambiguous or demanding. A very challenging one is molecular biology/biochemistry.

Biomolecules represent a huge collection of objects with individual structural, geometrical, qualitative and quantitative features. Although the feature representations are standardized to some extent — depending on the used structural formula (e. g. skeletal formula, Fischer projection, etc.) — learning to navigate in this knowledge domain using the graphical standards of the chemical nomenclature takes years (Brecher, 2006). In this project, I investigate the potential of infographics, graphical design, and game motivation for learning the features of a special group of biomolecules, the amino acids.

3.2 RELATED WORK

Many attempts have been made in scientific vulgarization to educate the public through card games. From the crowdsourced "Phylo" trading card game that makes use of the wonderful and inspiring things that inform the notion of biodiversity (mammals, bacteria, etc.) (Ng and Tan, 2010) to the chemistry card game "Molecules", which turns

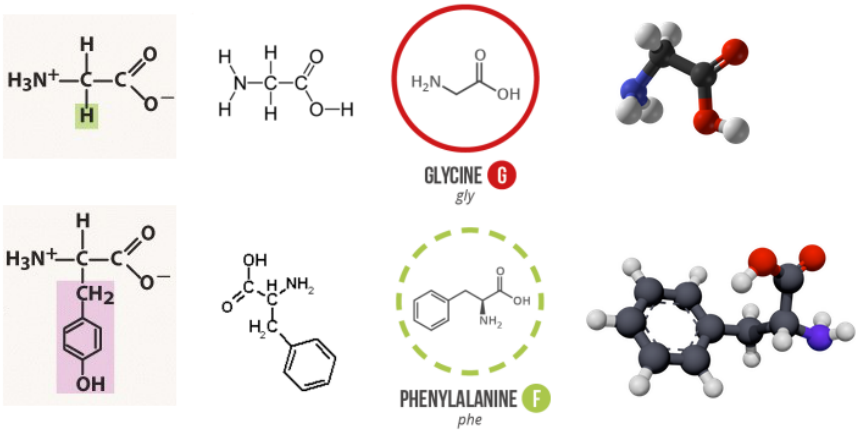


Figure 3.1: Current specialist representational forms and their shortcomings for less expert target audiences. Two amino acids were considered: glycine and phenylalanine. These representations were taken from Wikimedia Creative Commons and Compound Interest. The representations range from the Fischer representation (left) to the cyclohexane conformation (right). The differences between the molecules are easy to spot for the trained eye of a specialist. Yet, the first representation on the left is the clearest for less target audiences. This is due to simply highlighting differences (i. e. the side chain) and leaving the redundant part (i. e. common skeleton) in the background. All the other different representations are correct yet provide no means for easy recollection or guide the reader's eye and retain their attention.

learning atomic bonds into a competitive and fun task by building various compounds such as water or carbon dioxide (Dulek, 2015). In 2014, compound interest created a guide to the twenty common amino acids. This chart presents each molecule in a circle using a chart key which corresponds to each of the biomolecule categories (essential, acidic, etc.) with a minimal visual encoding. Most of the works did include the chemical formulas with no regard to emphasis on common and/or individual features (Figure 3.1). As a proof of concept, I propose a mnemonic card game based on creative design to aid memory retain amino acids. They have different features, which are often shared among more than one amino acid. An intuitive system to code these features into shapes, colors, and textures was developed. This will assist our abilities in interpreting visual stimuli, recognizing such features, grouping them, noting relationships, and ultimately memorizing the structural formulas (Bitterman, 1965). This approach could serve as a teaching tool for subject-oriented card game designs so to retain relevant information by using perceptual memory and fun.

3.3 METHODOLOGY

The collection of the data was performed through a formal research effort, i. e. looking up valid data from trustworthy sources. The data comprises: the amino acid names and properties, the different groups they can be put in based on their physical and chemical properties, and the structural formulas (*Römpp's Chemistry Lexicon*). The amino acids properties are: (a) molar mass [g/mol]: the given mass of a compound divided by its amount, (b) isoelectric point [no unit]: pH at which a molecule is neutral or does not migrate in an electric field, (c) solubility in water at 20°C [g/L]: ability of a solute in g to be dissolved in a liter of solvent, and (d) frequency in proteins (%): as reported for vertebrates in the Protein Data Bank (BERMAN et al., 2000). Those properties were chosen as attributes since they best reflect the chemistry of an amino acid and help determine its state given a certain environment. The unspecified stereochemistry was selected as an appropriate representation for each amino acid. The representations of the amino acids were documented and validated from previous biochemistry knowledge (NELSON, LEHNINGER, and COX, 2008). The designs "molecule" logo (on the back of each card) and "play" symbol (on the rules card) are courtesy of Ed Harrison (*The Noun Project*).

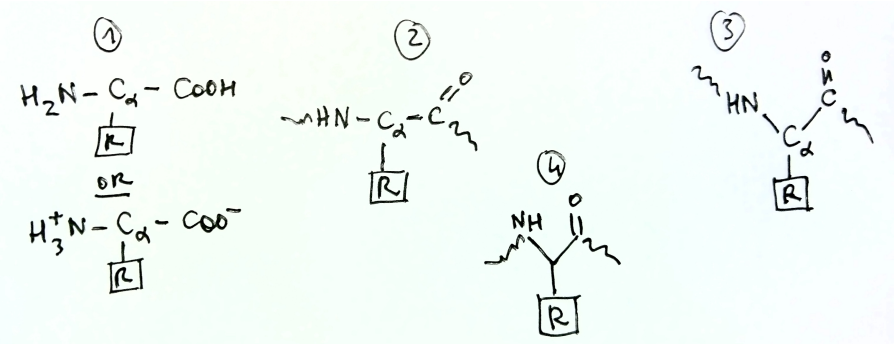
3.4 THE PROBLEM

As aforementioned, the basic problem is to render complex information easily accessible to the public and to ease its memorization. In this project, the information is the chemical structure of amino acids and their physicochemical properties. The major constraints affecting the design and development of the project were: clarifying pre-existing chemical formulas without altering the core information, finding ways to reinforce learning through visual encoding, creating a static (i. e. non-interactive) but concise visualization while working with the limited amount of card space.

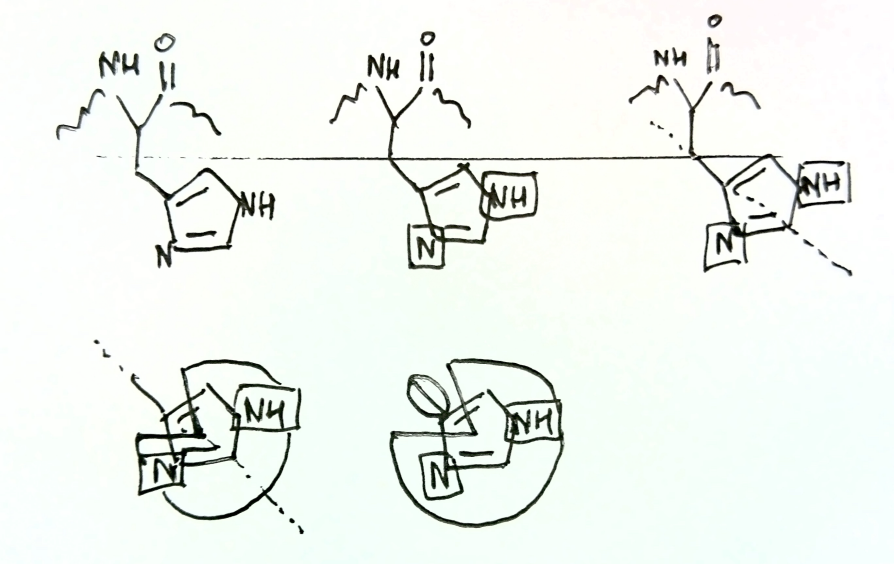
3.5 THE SOLUTION

The main purpose of this project was to create a teaching tool to better retain relevant information by using perceptual memory and fun. The targeted subject was a special kind of biomolecules, the amino acids. The design process could be applied to design other subject-oriented card games. The target audience comprises students, laboratory personnel, or any person in the public showing interest or desiring to know more about amino acids.

The proposed solution employs redundant visual encoding, stylized differences between the molecules, and interaction through gameplay to better learn the card's properties and categories. To memo-



(a) Sketches of the major iterations for the adopted representation



(b) Sketches of the histidine (H) amino acid

Figure 3.2: Sketches for the design process of the main molecule representation and an example amino acid (histidine) as a particular example. The redundant part of the molecule (i. e. common skeleton) is shared between the amino acids. (a) showcases these iterations. The squared R group is for the side chain. (1) represents the positively charged parts of the common skeleton (i. e. amino and carboxyl groups). (2) depicts the simplification and the adoption of the snake-like shapes using the unspecified stereochemistry representation. (3) portrays the process of finding a correct representation. (4) represents the final designs that were adopted: the snake-like shapes are in a correct position, symmetrical as is the Y-shape of the common skeleton. In (b), the example of the histidine molecule is depicted. The two nitrogen atoms are intrinsically symmetrical to the axis of the first carbon bond in the side chain (axis is depicted as a the dotted line). This permits to use that key feature and create a salient object to represent the side chain as a whole.

rize the structural formulas, the molecular features are encoded into shapes, colors, and textures. While researching the possible struc-

tural formula representations, I came across multiple solutions. These ranged from, but are not limited to, the molecular surface of the side chain, representations of the covalent radii of its atoms (HEYROVSKA, 2008), to the unspecified stereochemistry representation. Next, an appropriate design to help perceive differences among the amino acids was found. Then, the gameplay rules were articulated in a simple way. Lastly, the whole game was printed and tested, and its content was made openly available for download.

3.5.1 *Process for developing the solution*

The solution was shaped through a series of iterations, from sketching possible representations to the finally adopted design. Major iterations are reported in Figure 3.2. Using the example of the histidine side chain, I showcase how emphasis is brought to the side chain and de-emphasis is introduced to the common part of the molecule (see Figure 3.3). The finished design is present in Figure 3.5(a).

3.5.2 *Visual encoding*

First, I detail the encoding of the amino acids, as reported in Figure 3.3:

(1) Amino acids groups. Three main classifications exist: one that targets whether an amino acid is essential and two others that depend on the side chain structure (i. e. where differences occur). There exists multiple ways to group amino acids based on the side-chain. To adopt a compact grouping and support gameplay mechanics, four categories were selected: acidic, basic, polar and non-polar. The groups were visually represented by nominal colors and glyphs: blue — circle with minus sign, red — circle with plus sign, purple — empty circle and green — full circle, respectively. Saturated hues of these colors were chosen for a more vibrant card set (*Categorical Colours*). Each group has a corresponding category card which explains the main physicochemical properties of the grouped amino acids (see Figure 3.5).

(2) Amino acids name encoding. They were reported at the top of each card: the full name, the three letters code and the 1 letter code (example of lysine, Lys, K) (DAYHOFF, 1965).

(3) Stylized differences: emphasis and de-emphasis. The former is given to changing parts of the molecule (in the foreground). On the contrary, the latter is employed for the common part of the molecule in the background (Figure 3.3). De-emphasis was brought by layering wave-like lines on top of the common

part. It created the effect of a texture. Only one among twenty amino acids (proline) is exempt of the common part and hence only emphasis is used. In the case of emphasis, the visual encoding is more elaborate. Emphasis itself is split into two: The first tackled the abstract gray shapes that help memorize the side chains based on the amount of carbon bonds. The use of light and dark gray helps perceive differences by using the gestalt principles of similarity and proximity (e.g. axial/central symmetry in Figure 3.4) (Arnheim, 1949, 1956; Chang, Nesbitt, and Wilkins, 2007). Change in luminance reflects an asymmetry. The second addressed the presence of peculiar atoms (i. e. as sulfur — S and nitrogen — N) by using color coding and shape (Figure 3.3). They are highlighted using a yellow circle and teal blue rectangles, respectively (Figure 3.4 and Figure 3.3).

(4) Amino acid properties. The attributes: molar mass, isoelectric point, solubility, and frequency are represented by symbols: a scale, an electric sign, a container (i. e. erlenmeyer flask), and a pie chart, respectively.

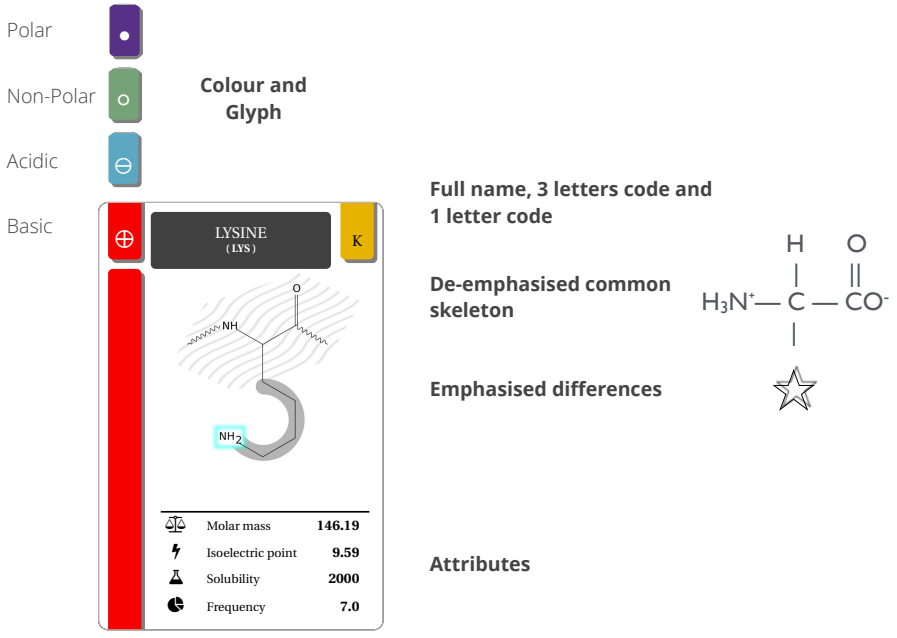


Figure 3.3: The major four steps to visually encode each amino acid card. Color and glyph category encoding, the name encoding, the stylized differences encoded (emphasis and de-emphasis), and the attributes.

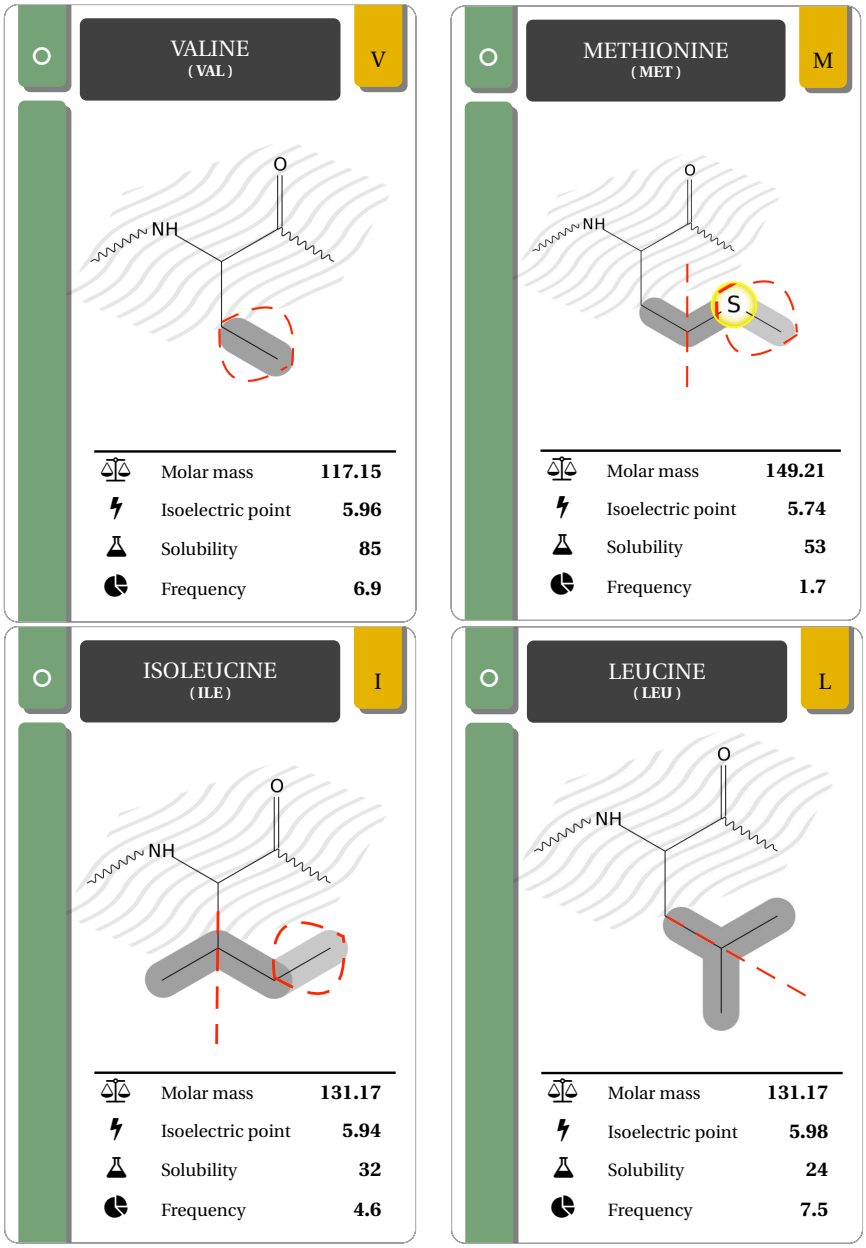
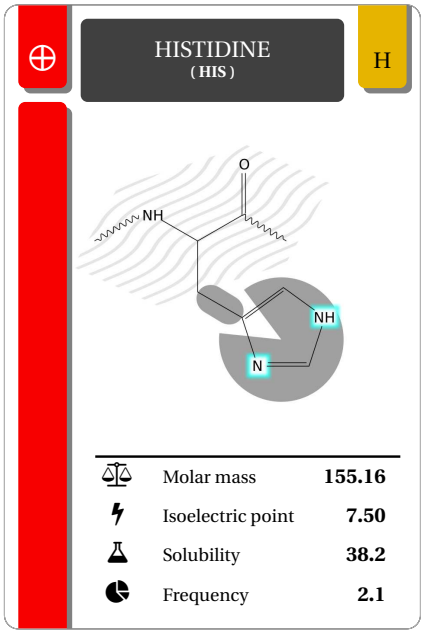
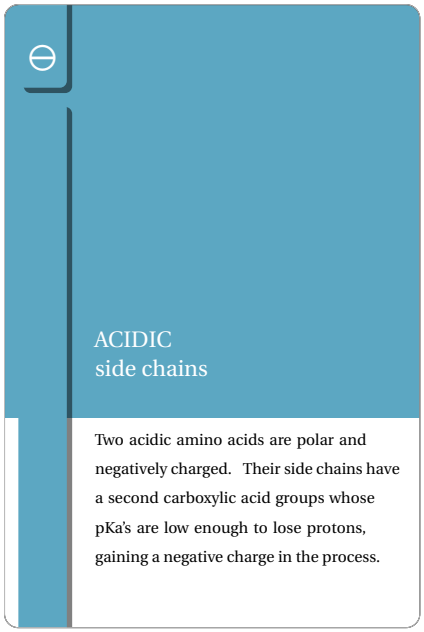


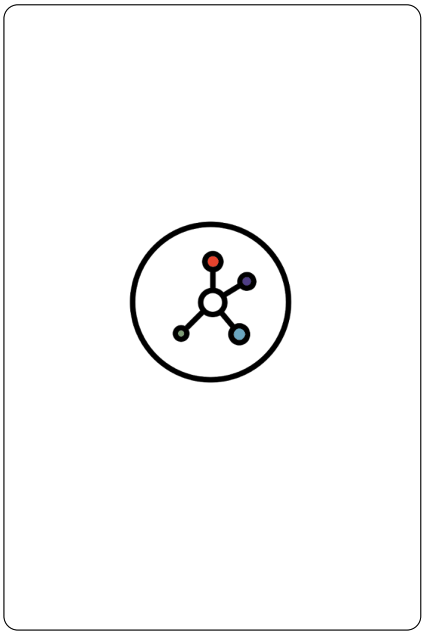
Figure 3.4: Stylized differences explained for a subset of amino acids pertaining to the non-polar group. De-emphasized common skeleton in the background using wave-like lines. Emphasized differences in the foreground are depicted in luminance (i. e. two different grays, light and dark). This emphasis depends on the amount of carbon bonds. Whereas light and dark grays are chosen to perceive symmetries (i. e. axial symmetry: dotted red line), asymmetries (denoted in a red polygon) and special atoms (e.g. zoomed-in and encircled sulfur (S) atom).



(a) Example amino acid card



(b) Example category card



(c) Back design of each card

(d) Rules card

Figure 3.5: Example playing cards included in the cards game. (a) showcases the histidine (H) amino acid with its respective formula, the category color and symbol (left and top left of card) which is in this case the basic category (or positively charged). The lower part of the card presents four numerical values. The molar moss (g/mol), the isolectric point or pI, the solubility in water (g/L), and the frequency in vertebrates (%) are represented as the symbol of a scale, a lightning, a flask, and a pie chart, respectively. (b) is a category card, it explicates the amino acids properties pertaining to this category. (c) depicts the back design for each card. Logo adapted with the colors of the four categories, courtesy of Ed Harrison (CC). (d) briefly lists the important rules.

Secondly, the cards that aid the gameplay also required their own design. The four category cards were designed using the same template of each amino acid card. The attributes of each card were replaced by an explanation of the category (Figure 3.5.b). The last card is the rules card, making the game a total of 25 cards. It depicts a brief explanation of the attributes and how to play. It is presented in Figure 3.5.

For two players the rules can be formulated in a simple manner: One player defines a challenge scope as either all the opponent's cards, one category, or one amino acid. For a chosen challenge scope, one player challenges his or her opponent on an attribute of a card from the cards the challenged player holds. The person with the highest attribute requests the card of the opposing player. This card is required to be the highest in hand. To win the game, a player must bank at least 9 cards by category (or color). In the case of 2 players, banking 10 cards is a requirement.

3.5.3 *Key milestones*

All of the aforementioned sections and development steps were realized. Main decisions ranged from but were not limited to: preserving the initial chemical formula without losing too much information, rendering the game as user friendly as possible, and making all the content openly available to the public.

3.6 RESULTS

The solution in addressing the initial problem using playing cards as material proved to be successful. The game was made openly available to the public for download, distribution, and printing (i. e. full card set and gameplay rules).

An informal qualitative assessment and usability test was carried out on a group of 10 students. In most cases, the players had a vague memory of the amino acids structures and the card game worked out well in helping them remember the main features and properties. The card attributes helped them place the molecules in their minds as being more or less identical. Coupled with the desire to win, players knew why they lost a challenge and which card is better. By using the actual molecule properties as card attributes, players better navigated in the space of values. The standard card size and chosen paper thickness (0.8 mm) were reported as favorable.

3.7 DISCUSSION

In the broader category of developed card games, this is the first attempt to go one step further in the design phase and abstract shapes

from chemical structures. These shapes aided perceptual memory to retain relevant features and structures that are intrinsic to amino acids. The most important lessons were: 1. redundancy and consistency in visual encoding helps to better learn the relevant categories and molecular features, 2. the interaction through gameplay helps to memorize the actual properties and navigate the values, and 3. the fun factor is important to enjoy and learn.

This project provides a viable solution for non-interactive, task-oriented visualization, yet faces a couple of limitations. Firstly, if the intent is to solely memorize the formulas without playing, the cognitive load or mental effort in learners could be consequent. Compared to the classical approach, where only formula representations are used, the card game solution provided a reduced mental effort for memorizing 20 amino acids. Secondly, if the card game is indeed used as intended, early learners might face difficulties such as remembering the different features. This suggests for further research, where "eye candy" could be used to undermine cognitive load, to attract and entertain learners. Additionally, a large-scale evaluation of the card game could prove useful.

In the context of this thesis, this project proves that for certain use-cases a non-interactive visualization can be preferable. Card games have a long tradition and their public familiarity lowers the learning curve for this visualization. The fun factor aids both the process of memorizing as well as overcoming the initial reluctance to start learning the something new. Furthermore, the physical print version of the cards makes it possible to take them anywhere and use them anytime, in contrast to a computer based visualization. Admittedly, amino acids are a comparably simple type of data and only the most important physicochemical properties were selected for the task. Often, both data and task are much more complicated, rendering the use of a computer based interface mandatory, which will be the topics of part II and III of this thesis.

Part II

PRESYMPTOMATIC DIAGNOSIS OF ORAL CANCER

The second part of this thesis discusses a specific use case of data mining and visualization in the medical field, i. e. the study of tumor development. The fourth chapter details the biological and medical background for this project and introduces the data at hand. The fifth chapter defines the problem statement and presents *ddPCRclust*, an R package for the automated analysis of multiplexed, non-orthogonal droplet digital PCR data. The sixth chapter follows with an accompanying visual interface *ddPCRvis*, which provides access to the algorithm through a web browser, enabling the user to interactively filter data and adjust parameters, as well as view and modify results. The seventh chapter compares the automated clustering approach to manual annotation by experts and discusses the results.

4

BIOLOGICAL BACKGROUND

As the name suggests, a task oriented visualization focuses on a specific task at hand. In this chapter, I introduce the biological background of this task: the microbiology of cancer. I explain, how the data in this project is produced and why it is necessary to develop a specific algorithm to analyze it. I limit my scope to cells with a distinct nucleus, i. e. eukaryotes.

4.1 DIVERSITY AND DYNAMICS OF CANCER

Cancer is undoubtedly a global public health problem and, despite the efforts made, continues to affect and kill a huge number of people without distinction. In 2012, about 14 million new cases of cancer occurred globally, accounting for about 8 million or 14.6 % of human deaths. One of the main reasons why cancer is difficult to treat is its diversity. The term cancer actually encompasses a huge group of diseases, which all involve abnormal cell growth with the potential to invade or spread to other parts of the body, but can have very different causes. For humans, there are over 100 known types of cancer (Stewart and Wild, 2014).

This number still appears small compared to the fact that the adult human is composed of approximately 10^{15} cells, many of which are required to divide and differentiate in order to form organs and tissues. There is a sensitive balance between the generation of new cells, called *proliferation*, and the natural death of cells, called *apoptosis*. In case of cancer, this balance shifts towards uncontrolled proliferation, causing new and abnormal growth of tissue. This behavior is generally defined as a *neoplasm*, with cancer falling into the category of malignant neoplastic diseases (Bertram, 2000).

What causes cells to grow and spread abnormally, harming their own body in the process? The answer lies in the genome, the instructions facilitating growth, development, functioning, and reproduction of all known living organisms. This information is stored in macromolecules known as deoxyribonucleic acid (DNA), which consist of four nitrogen-containing nucleobases — cytosine (C), guanine (G), adenine (A), and thymine (T) — plus a backbone out of deoxyribose and phosphate. Adenine and thymine, as well as guanine and cytosine form a hydrogen bond, and are hence called base pairs. Together with the sugar-phosphate backbone they form nucleotides, the building blocks of DNA. Many of them in a row create a double helix

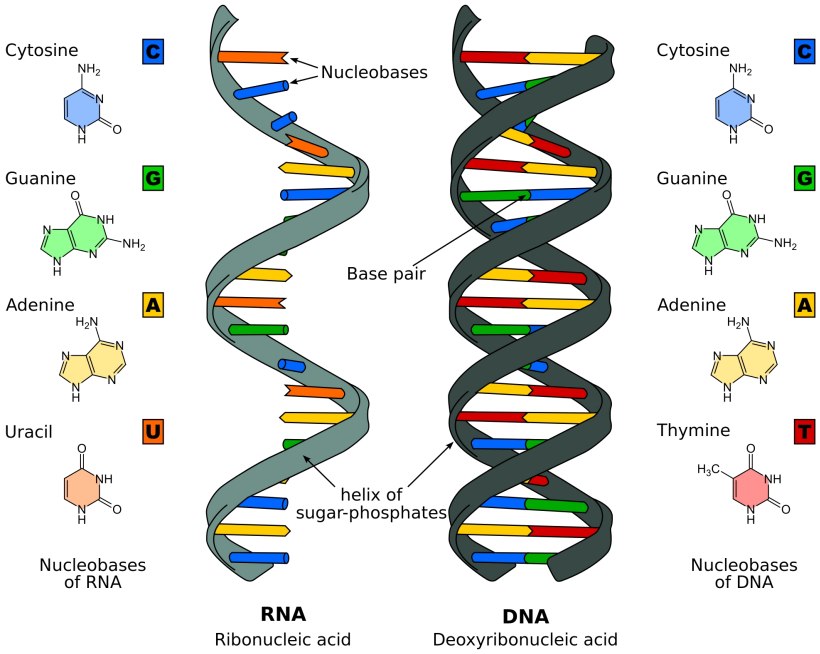


Figure 4.1: The difference between RNA and DNA. RNA is a single stranded molecule consisting of the nucleobases cytosine (C), guanine (G), adenine (A), and uracil (U). DNA is a double stranded molecule consisting of the nucleobases cytosine (C), guanine (G), adenine (A), and thymine (T).
Reproduced from: `https://commons.wikimedia.org/wiki/File:Difference_DNA_RNA-EN.svg`.

structure, as seen in Figure 4.1, which gives the long DNA molecules their stability.

Whenever genetic information needs to be accessed, DNA is transcribed into ribonucleic acid (RNA). RNA molecules are similar to DNA, except the complementary base to adenine is uracil (U), which is an unmethylated form of thymine, the backbone contains ribose instead of deoxyribose, and, most importantly, RNA is usually a single-stranded molecule (Figure 4.1).

DNA molecules within a eukaryotic cell are organized in chromosomes, which are further enclosed by a membrane, forming the nucleus. Humans have 46 chromosomes, consisting of 23 pairs — one from each parent — whilst some animals can have up to 268 chromosomes (*Agrodiaetus shahrami*). Each chromosome is duplicated during a cell division, except for gametogenesis, where each gamete only receives a single copy of each chromosome. During the duplication, the DNA molecules may undergo several recombination events. This is favorable for genetic diversity in gametogenesis, but incorrect recombination may lead to chromosomal abnormalities. Mistakes during gametogenesis can lead to developmental diseases (e. g. Down syndrome, Huntington's disease) or increased risk factors for other diseases (e. g. diabetes, cancer). However, mistakes during cell divi-

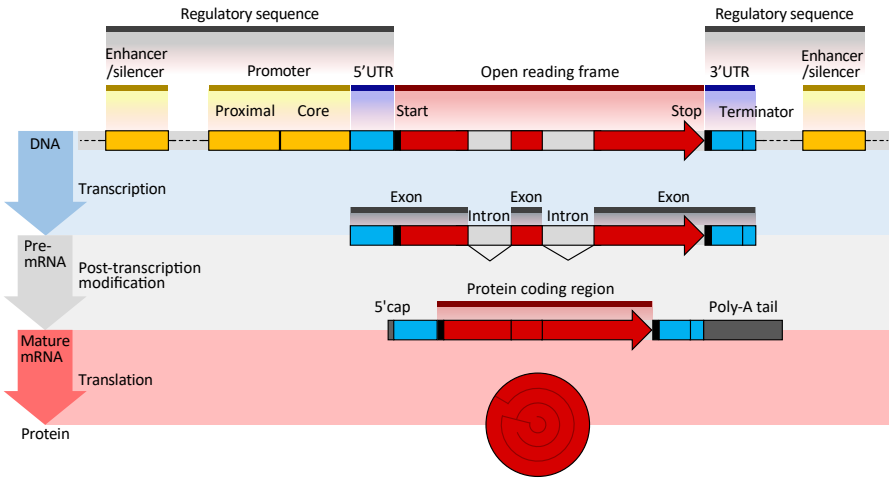


Figure 4.2: The structure of a eukaryotic protein-coding gene. Regulatory sequence controls when and where expression occurs for the protein coding region (red). Promoter and enhancer regions (yellow) regulate the transcription of the gene into a pre-mRNA which is modified to add a 5′ cap and poly-A tail (grey) and remove introns. The mRNA 5′ and 3′ untranslated regions (blue) regulate translation into the final protein product. Vertical: Three steps from DNA (blue) to protein (red).
Reproduced from: `https://commons.wikimedia.org/wiki/File:Gene_structure_eukaryote_2_annotated.svg`

sion are not limited to gametogenesis and can happen anywhere in the body with severe consequences, such as the death of the cell or genetic diseases like cancer (see Section 4.1.1).

Certain regions within the chromosomes are known as genes. The definition for a gene has changed a lot throughout the history of molecular biology and is now simplified to "a molecular unit of heredity" (Slack, 2014). It typically consists of an open reading frame (ORF), as well as all corresponding regulatory elements, as shown in Figure 4.2. The ORF is first transcribed into a messenger RNA (mRNA), which can subsequently undergo post-transcriptional modifications such as splicing, before finally being translated into a protein.

Transcription is happening in a cell's nucleus, where the chromosomes are located, but translation is performed by ribosomes outside the nucleus. During the translation, the RNA sequence is decoded into a sequence of amino acids, which in turn form a protein. The key for decoding RNA sequence is the genetic code. Each amino acid is encoded by a triplet of nucleotides called codon. The number of possible combinations for four different nucleotides is $4^3 = 64$, however most eukaryotes only use 20 proteinogenic amino acids. Even adding the three stop codons, which terminate the transcription, the number is still a lot smaller. Thus, many amino acids have more than one triplet coding for them. This way not every change in the DNA sequence is passed along to the protein.

Proteins are large molecules consisting of one or more long chains of amino acids. Their function is determined by their three-dimensional structure, which is dictated by the sequence of amino acids. Many proteins fold into their native state unassisted once the translation is complete, simply through the chemical properties of their amino acids. Others require the aid of other proteins to fold correctly. This is one example of a vast array of functions that proteins perform within organisms, including catalyzing metabolic reactions, DNA replication, responding to stimuli, and transporting molecules from one location to another.

This transport of information from DNA to protein is the very foundation of life and involves many regulatory elements. Most of them can be found in the regions of the DNA that do not belong to ORFs and are thus not translated to proteins, although they are often in close proximity. The perception of these regions has also changed over time, from being called "junk DNA" (Ohno, 1972) to the more neutral term non-coding DNA, which is now known to not only contain many regulatory elements, but also scaffold attachment regions, origins of DNA replication, centromeres and telomeres, and more. If any alterations occur in the DNA molecule, whether inside an ORF or not, they can be passed along this chain of information. These alterations are called mutations and they can be both beneficial or harmful to an organism.

4.1.1 *Mutations*

All neoplastic diseases are caused by mutations to the cell's genome. A mutation is defined as a permanent alteration to the DNA of an organism. This damage can be the result of endogenous processes such as errors in replication of DNA, the intrinsic chemical instability of certain DNA bases, or from "attack" by free radicals generated during metabolism. DNA damage can also result from interactions with exogenous agents such as ionizing radiation, UV radiation, and chemical agents. Three different types of mutations can be distinguished:

- The smallest possible alteration is a simple substitution, where one nucleotide is exchanged for another. Such a point mutation can be silent, the changed base triplet codes for the same or a sufficiently similar amino acid. However, it may also code for a different amino acid or a stop codon, which can truncate the resulting protein, altering it's shape or function, or completely inactivating it. In the context of comparing two or more genomic sequences, this category of mutations is also defined as single-nucleotide polymorphism (SNP).

- Insertions of extra nucleotides into the DNA or deletions of one or more nucleotides from the DNA are typically grouped together under the term *indel*, since they have similar effects. If

their length is not a multiple of three, they will produce a frameshift, i. e. shifting the reading frame for all following nucleotide triplets, significantly altering the gene product.

- Large scale chromosomal mutations include duplications or deletions of large regions or an entire chromosome, as well as translocations (interchange of genetic parts between chromosomes) or inversions (reversing the orientation of a chromosomal segment).

A cell has a collection of processes to identify and correct any damage to its DNA molecules (SANCAR et al., 2004). However, the more mutations occur, the more severe those mutations are, the more difficult it becomes to repair them. Any mutations that occur in genes responsible for maintaining genomic integrity also facilitate the acquisition of additional mutations. Thus, mutagenic substances like some parts of tobacco smoke increase the rate at which mutations occur significantly, in turn increasing the chance to suffer from neoplastic diseases. Hence these mutagenic substances are also called carcinogens.

4.1.2 *Copy number aberrations*

Any indel or large scale chromosomal mutation can lead to the loss or gain of certain regions. Naturally, any genes or regulatory elements located in these regions will be lost or gained as well. In general, this concept is called copy number variation (CNV), however in the context of somatic CNVs, these are often referred to as copy number aberration (CNA). CNAs are extremely common in neoplastic diseases and play an important role in their progression (HANAHAN and WEINBERG, 2011). In cancer, we can typically distinguish two types of CNAs: Those which increase the activity of genes that facilitate cell proliferation — this class of genes are called oncogenes. And those which inactivate gene function in the case of genes responsible for regulating proliferation and inducing apoptosis — this class of genes is called tumor suppressor genes. Discovering oncogenes and tumor suppressor genes is crucial in understanding the underlying mechanics of the cellular defects that cause cancer and suggesting potential therapeutic strategies (BEROUKHIM et al., 2010).

4.1.3 *Allelic imbalance*

Most mammals, including humans, are diploid, meaning their cells contain two complete sets of chromosomes — one from each parent. Accordingly, each gene should be present exactly twice in each cell, with each variant being defined as one allele. If both alleles of a gene are the same, they and the organism are homozygous with respect

to that gene. If the alleles are different, they and the organism are heterozygous with respect to that gene. CNVs can occur on one of the chromosomes, leading to two alleles of a given gene being expressed at different levels in a given cell. This is defined as allelic imbalance (AI). If one of the alleles is completely lost, it is referred to as loss of heterozygosity (LOH).

Advances in the molecular-genetic analysis of cancer cell genomes have provided evidence of ongoing genomic instability during tumor progression. AIs have been found in many cancers, for example in breast cancer (Cleton-Jansen et al., 1994), prostate cancer (Cher et al., 1994), colorectal cancer (Halling et al., 1999), or oral cancer (Partridge et al., 1998). However, some chromosomal loci seem to be more commonly affected by AIs than others. This indicates that such sites are likely to harbor genes whose alteration favors neoplastic progression, making them especially interesting as targets for detection of early stages or precursors of cancer (Hanahan and Weinberg, 2011).

4.1.4 *Cancer precursors*

By their very nature, neoplastic diseases evolve rapidly over time. The genetic instability during tumor progression provides the cancer with the *genetic diversity* required for natural selection and enables the extensive *phenotypic diversity* that is frequently observed among patients. It also causes the formation of subpopulations as the tumor progresses, making targeted treatment extremely difficult. Hence undirected chemotherapy with cytotoxic agents remains a standard treatment for a vast majority of cancer patients.

Many forms of cancer are known to form from precursor states, before evolving into a tumor. Identification of these precursors, or precancers, is important to elucidate critical early steps in cancer development, to determine targets for chemopreventive agents, and to identify easier treatable precancers destined to progress to an invasive disease. It has been shown that AIs play a crucial role in progression from precancerous stages to tumors (Hanahan and Weinberg, 2011; Larson et al., 2002), thus presented an interesting target for the detection of precancers.

4.1.5 *Oral cancer*

Oral cancer has a major impact worldwide, accounting for 274 000 new cases and 145 000 deaths each year, making it sixth most common cancer (Ferlay et al., 2015). The 5-year survival rates, which range from 30–60 %, are among the worst of all cancer types. In most cases, oral cancer is treated surgically, while radiation therapy and chemotherapy serve as adjuvant treatments. Even when successful,

the results are often diminished quality of life, impaired function and disfigurement.

There are several types of oral cancers, but around 90 % are squamous cell carcinomas (SCCs). SCCs are known to develop from cancer precursor stages, so called oral premalignant lesions (OPLs). These lesions have a high prevalence amongst smokers and alcoholics, but can also occur due to poor oral hygiene, irritation caused by ill-fitting dentures and other rough surfaces on the teeth, poor nutrition, and some chronic infections caused by fungi, bacteria or viruses (SRINIVASPRASAD et al., 2015).

In early stages, oral cancer might go unnoticed. Lesions are often without pain and only slight physical changes. Later stages involve symptoms like bleeding sores; lumps or thickening of the skin; pain in tongue, jaw, or throat; airway obstruction or loose teeth. Advanced oral cancer is also known to metastasize (i. e. spread) through the lymphatic system into lymph nodes, liver, and kidneys (MYERS, 2009).

4.2 DETECTING ALLELIC IMBALANCES

Biopsies can be performed on any equivocal tissue in order to detect the cancer at its earliest stage. However, since OPLs and very early stages of oral cancer show little symptoms, many people will not consult a physician at that point. Even if they do, it has been shown that biopsies of OPLs are often not reliable (HOLMSTRUP et al., 2007). Thus, a more sensitive and reliable approach based directly on genetic information instead of optical observations of the tissue needed to be sought. ZHANG et al. (2012) identified and validated AIs, such as LOHs, as risk predictors for progression from OPLs to oral cancer. Discovering these AIs in an early stage can significantly increase the chances of successful treatment and avoid relapses.

4.2.1 *Amplifying DNA*

In order to detect AIs, DNA of equivocal cells needs to be analyzed or sequenced. To do so, most technologies first require an amplification step to obtain a sufficiently high number of copies of the sequence of interest. The *de facto* standard for amplifying DNA molecules is polymerase chain reaction (PCR). The general procedure for a PCR reaction is as follows.

1. The DNA molecule needs to be denatured, i.e. the two strands are being separated by heating the sample to 94–98 °C.

2. The temperature is lowered to a value that allows the primers of this reaction to bind to the DNA template. Primers are specifically designed oligonucleotides, typically ranging from 18 to 30 base pairs, which are complementary to the 5′- or 3′-end of

the DNA sequence of interest, respectively. This step is called annealing and lasts about 20–40 seconds.

3. A DNA polymerase synthesizes a new DNA strand complementary to the DNA template strand by adding free deoxynucleotide triphosphates from the reaction mixture that are complementary to the template in the 5′-to-3′ direction, starting at the respective primers. The temperature at this elongation step depends on the DNA polymerase used and the duration on the length of the template.

The processes of denaturation, annealing and elongation constitute a single cycle. This cycle is repeated, until the desired amplification has been achieved. Each cycle the number of copies of the DNA template is doubled, so after n steps the number of copies theoretically equals 2^n. Practically, experimental variance and amplification bias will affect this result, making PCR reactions difficult to compare (Acinas et al., 2005).

4.2.2 *Digital PCR*

Detection and quantification of specific nucleic acid sequences using PCR has a long history in molecular biology (Southern, 1975; Thomas, 1980). Soon after its presentation by Heid et al. (1996), real-time PCR became the standard for quantification of nucleic acids. The quantitative information is obtained from the cycle threshold (CT), a point on the analogue fluorescence curve where the signal increases above background. With the knowledge that the template is approximately doubled each cycle, it is possible to estimate its concentration. However, external calibrators or normalization to endogenous controls are required and imperfect amplification efficiencies affect CT values, which in-turn limit the accuracy of this technique for absolute quantification. In digital PCR, the target DNA is distributed across a large number of partitions and the reaction is carried out in each partition individually. Due to this dilution, some partitions will have no template and others will have one or more template copies present. The PCR reaction is then carried out until its plateau phase, eliminating amplification efficiency bias (see Section 4.2.1). Partitions containing one or more templates yield positive end-points, whereas those without template remain negative. Using Poisson's law of small numbers, the actual number of template DNA molecules present can be derived from the fraction of positive end-point reactions, according to Equation 4.1,

$$\lambda = -\ln(1 - p) \tag{4.1}$$

where λ is the average number of template DNA molecules per replicate reaction and p is the fraction of positive end-point reactions.

From λ, together with the volume of each replicate PCR and the total number of replicates analyzed, an estimate of the absolute target DNA concentration can be calculated.

4.2.3 *Droplet digital PCR*

In the beginning, digital PCR arrays only offered hundreds of partitions, limiting the dynamic range of quantification (DUBE, QIN, and RAMAKRISHNAN, 2008). However, a more recent protocol by HINDSON et al. (2011) describes a variant called droplet digital PCR (ddPCR), which significantly increases the number of partitions while lowering the experimental costs, providing a boost to the technology (Figure 4.3).

ddPCR is an emerging technology for detection and quantification of nucleic acids. In contrast to other digital PCR approaches, it utilizes a water-oil emulsion droplet system to partition the template DNA molecules. Each droplet serves as a compartment for a PCR reaction, just as individual test tubes or wells in a plate, but on a smaller scale. This system enables the partitioning into up to 20 000 nanoliter-sized droplets, significantly increasing the dynamic range for detecting changes in DNA quantity such as AIs.

A typical ddPCR workflow is presented in Figure 4.4: (a) Samples and droplet generation oil are loaded into an eight-channel droplet generator cartridge. (b) A vacuum is applied to the droplet well, which draws sample and oil through a flow-focusing nozzle where monodisperse 1 nL droplets are formed. In under 2 minutes, eight

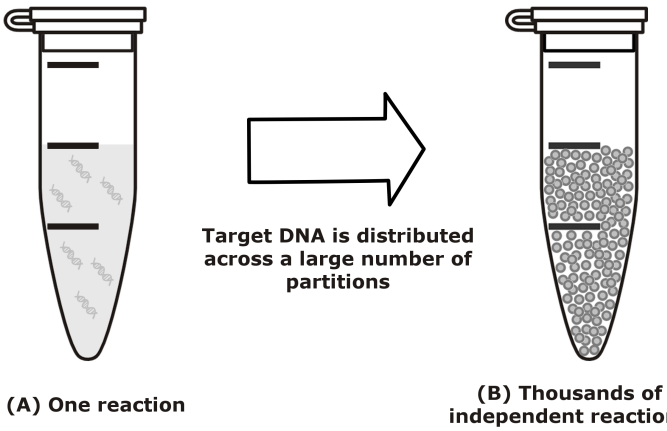
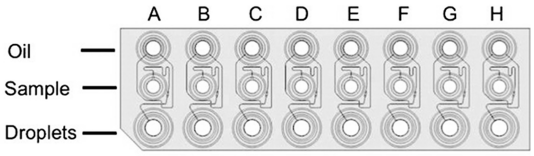
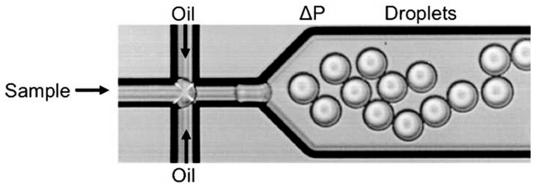


Figure 4.3: Sample partitioning is the key to droplet digital PCR. In traditional PCR, a single sample offers only a single measurement (A), but in droplet digital PCR, the sample is partitioned into 20 000 nanoliter-sized droplets (C). This partitioning enables the measurement of thousands of independent amplification events within a single sample.

(a) Load samples and oil into disposable droplet generator cartridge



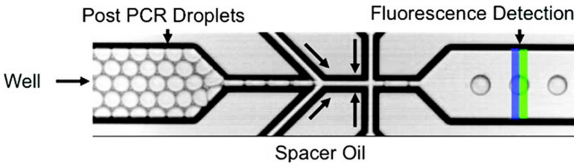
(b) Generate droplets



(c) Transfer droplets to 96-well PCR plate

(d) Thermal cycle to end-point

(e) Read droplet flourescence



(f) Apply amplitude thresholds and calculate conctentrations

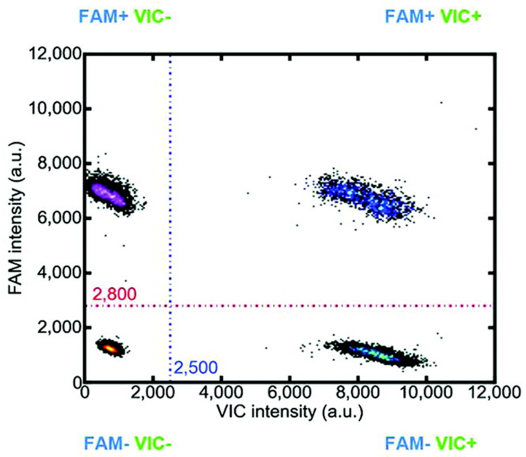


Figure 4.4: Overview over a typical droplet digital PCR workflow. (a) the sample is loaded, (b) the sample is partitioned into droplets, (c)–(d) sample is amplified using PCR, (e) fluorescence is measured for each droplet, (f) results need to be analyzed. (Adapted from Hindson et al., 2011).

samples are converted into eight sets of 20 000 droplets. (c) The surfactant-stabilized droplets are pipet transferred to a 96-well PCR plate. (d) Droplet PCR amplification to end-point (35–45 cycles) is performed

in a conventional thermal cycler. (e) The plate is loaded onto a reader which sips droplets from each well and streams them single-file past a two-color detector at the rate of ~1000 droplets per second. (f) The droplets are assigned as positive or negative based on their fluorescence amplitude. The number of positive and negative droplets in each channel is used to calculate the concentration of the target and reference DNA sequences (see Equation 4.1) and their Poisson based 95 % confidence intervals (HINDSON et al., 2011).

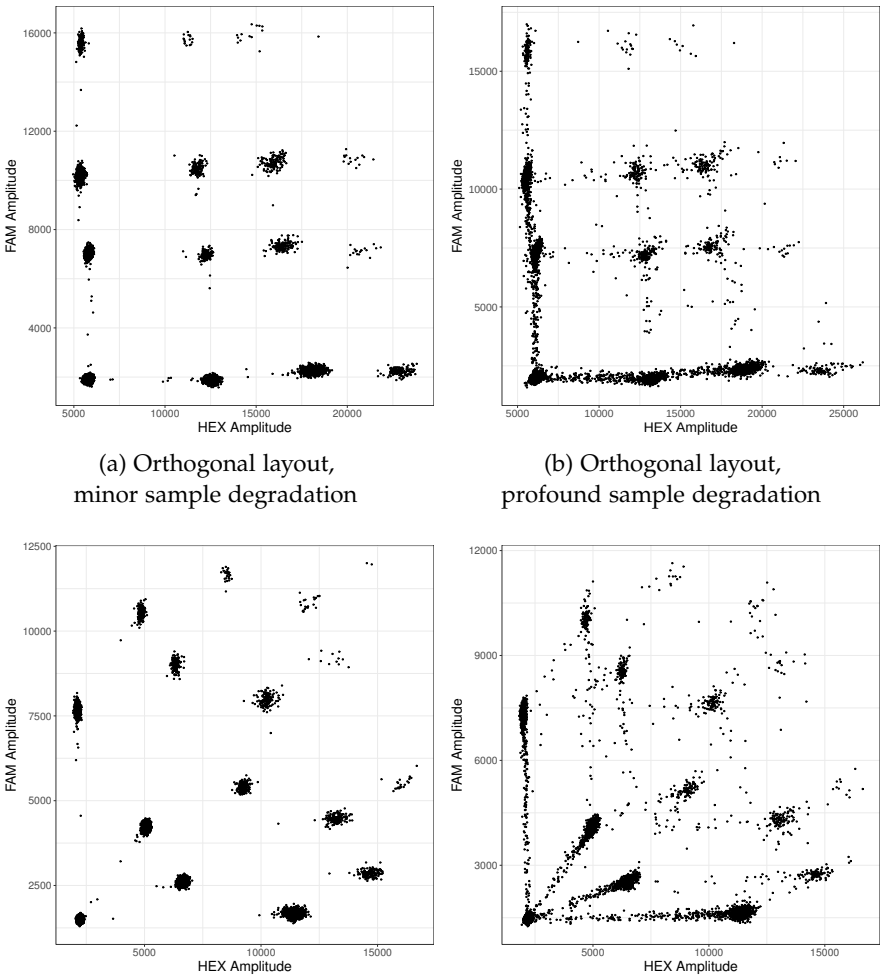
4.3 DETECTING CNAS WITH DDPCR

As aforementioned, during a ddPCR run, each genetic target is fluorescently labeled with a combination of two fluorophores (typically HEX and FAM), giving it a unique footprint in a two-dimensional space represented by the intensities per color channel. The position of each droplet within this space reveals how many and, more importantly, which genetic targets it contains. Thus, droplets that contain the same targets cluster together (see Figure 4.5). The number of positive droplets for each target determines its abundance, which can be used to detect CNAs in clinical samples.

Although the specifics of genome alteration vary dramatically between different tumor types, CNAs within the human genome are known to correlate with the development and progression of cancer (see Section 4.1.2). Quantifying CNAs has therefore become a fundamental part of oncology and inspired numerous research in this direction. However, their accurate detection by ddPCR presents a unique and considerably greater challenge. It requires the quantification of subtle changes in the abundances of genetic regions by comparing the corresponding abundances of specific biomarkers relative to the average ploidy of the tissue. If that tissue section is formalin-fixed paraffin-embedded (FFPE) and of a small size such that only a limited amount of DNA can be extracted, as is often the case with clinical samples, application of ddPCR to CNA determination becomes even more difficult.

The reason for this is that, in addition to the low quantity and quality of the DNA generally obtained, damage in the form of sequence alterations can further reduce the amplification efficiency. This results in droplets with their respective signal lying along a vector connecting two clusters in the ddPCR output, which can contain up to half of the droplets intrinsically belonging to the higher order cluster in case of profound sample degradation. This phenomenon is called *rain*, referring to the cloudy shapes of the clusters.

Another drawback is the limited throughput of ddPCR, since current generation ddPCR hardware only supports detection of two color channels, hence originally only providing means for a duplex reaction. A lot of effort has been made to improve the throughput by mul-



(a) Orthogonal layout, minor sample degradation

(b) Orthogonal layout, profound sample degradation

(c) Non-orthogonal layout, minor sample degradation

(d) Non-orthogonal layout, profound sample degradation

Figure 4.5: Examples for ddPCR data. In (a) and (c) all 16 clusters are present. In (b) and (d) the sample was partially degraded, causing formation of *rain* and disappearance of the higher order clusters. Non-orthogonal layout avoids overlap of clusters and *rain*.

tiplexing the reactions, i. e. running more than two targets at once. McDermott et al. (2013) have introduced a way to combine the two fluorophores to analyze a third target. Dobnik et al. (2016) proposed combining the two available colors to run four DNA templates at the same time, by doubling the amount of fluorophore used for the other two targets (see Figure 4.5 (a)).

Yet simply doubling the amount of fluorophore results in an orthogonal cluster layout. In clinical FFPE samples, rain is overlapping with other cluster centers, hence making it impossible to distinguish which droplet belongs to which cluster (see Figure 4.5 (b)). A recently published protocol by Hughesman et al. (2016) has further refined the multiplexed ddPCR methodology to be able to analyze four tar-

gets and use a non-orthogonal layout in order to avoid overlapping of clusters and rain (Figure 4.5 (c) & (d)).

However, dedicated software for the automated analysis of multiplexed ddPCR reactions aimed at detecting three or more targets is not yet available. The commercial QuantaSoft™ software, which handles the raw data from ddPCR reactions, does not support the detection of rain or non-orthogonal cluster layouts. In order to analyze a reaction, users have to manually draw boarders around each cluster (see Figure 4.6). Analyzing the data this way takes multiple hours to complete, hence presenting a major bottleneck. Furthermore, manual analysis has the usual disadvantages of subjectivity and non-reproducibility. In order to determine whether signs of cancerous progressions are apparent on a clinical scale, efficient algorithms need to be sought.

Figure 4.6: Manual analysis of ddPCR data. Using the commercial QuantaSoft™ software, manual analysis is only possible by manually drawing boarders around the clusters and labeling them by hand.

5 THE DDPCRCLUST PACKAGE

As presented in Chapter 4, data from ddPCR consists of a number of different clusters $c_1, \ldots, c_k$, which each contain droplets representing one or more genetic targets $t_1, \ldots, t_l$. The commercial QuantaSoft™ Software requires an orthogonal layout, rendering it unusable for clinical samples. Manual annotation is time consuming and hinders the analysis of ddPCR data. It presents a major bottleneck for the technology, which I address in this chapter, by presenting an an R package and associated interface (*ddPCRvis*) for automated analysis of multiplexed ddPCR samples. I first present related work in this field, then I introduce the problem statement and the methodology behind *ddPCRclust*. Parts of this chapter have been submitted as Brink, Meskas, and Brinkman (2018). The package is available under `https://github.com/bgbrink/ddPCRclust`.

5.1 RELATED WORK

To address the challenges presented by manual analysis of ddPCR data, several automated methods have been developed. *ddpcRquant* proposes estimating a threshold for gating by modeling the extreme values (Trypsteen et al., 2015). Two other R packages, *ddPCR* (Attali et al., 2016) and *twoddpcr* (Chiu et al., 2017), include graphical user interfaces (GUIs) built upon Shiny (Chang et al., 2017), a web application framework for R (R Core Team, 2017), providing easy access to their respective algorithms and allowing for manual correction of the analysis. However, analysis of multiplexed ddPCR reactions (i. e. reactions with more than two targets) is not supported by most tools, as presented in Table 5.1. Thus, I developed *ddPCRclust*, which supports both manual and automated analysis of non-orthogonal, multiplexed ddPCR reactions.

Table 5.1: Comparison of available tools for analysis of ddPCR data.

	ddpcRquant	ddpcr	twoddpcr	QuantaSoft	**ddPCRclust**
Manual gating	no	yes	yes	yes	**yes**
Automatic gating	yes	yes	yes	yes	**yes**
Targets supported	2	2	2	4	**4**
Rain supported	no	yes	yes	no	**yes**
Freely available	yes	yes	yes	no	**yes**

5.2 PROBLEM STATEMENT

During a ddPCR run, each genetic target is fluorescently labeled with a combination of two fluorophores (typically HEX and FAM), giving it a unique footprint in a two-dimensional space represented by the intensities per color channel. The position of each droplet within this space reveals, how many and, more importantly, which genetic targets it contains. Thus, droplets that contain the same targets cluster together (see Section 4.2.3).

Since one droplet can contain more than one target, the number of possible clusters depends on the number of targets. Following the laws of combinatorics, the number of ways to choose a subset of k elements, disregarding their order, from a set of n elements, is defined as the binomial coefficient $\binom{n}{k}$. Thus, summing up all the possible combinations to choose k elements out of t targets yields Equation 5.1,

$$\sum_{k=0}^{t} \binom{t}{k} = 2^t \tag{5.1}$$

with t being the number of targets in this reaction and k the possible number of targets per individual droplet.

However, the correct combination of targets for each droplet is only given implicitly by its fluorescence footprint. Let x be a single droplet signal. Each x consists of two features $x = (x_1, x_2)$. These features represent the fluorescence intensities measured by the ddPCR machine and I define the tuple as the fluorescence footprint of x.

Let $\mathcal{X} = \{x_1, \ldots, x_n\}$ be the set of all droplets (i. e. one data file) and therefore $x_i = (x_{i,1}, x_{i,2})$, with $i \in \{1, \ldots, n\}$. Let $\mathcal{L} = \{l_1, \ldots, l_n\}$ be the set of all labels for $\mathcal{X}$, with $l_i \in \{1, \ldots, 2^t\}$ according to Equation 5.1. Thus, the clustering problem can be defined as finding a function ρ that assigns each x to a label l, as presented in Equation 5.2.

$$\rho : \mathcal{X} \to \mathcal{L}, x \mapsto l \tag{5.2}$$

Let $\mathcal{T} = t_1, \ldots, t_l$ be the set of all targets. Each target t_j also has a specific fluorescence footprint $f_j = (f_{j,1}, f_{j,2})$ and $t_j \mapsto f_j$. This implies that in theory, the fluorescence footprint for each x can be defined as the sum of the fluorescence intensities of all its positive targets according to Equation 5.3.

$$\begin{aligned} x &= \sum_{j=1}^{l} f_j \\ fj &= \begin{cases} (f_{j,1}, f_{j,2}), & \text{if target } t_j \text{ positive} \\ 0, & \text{else} \end{cases} \end{aligned} \tag{5.3}$$

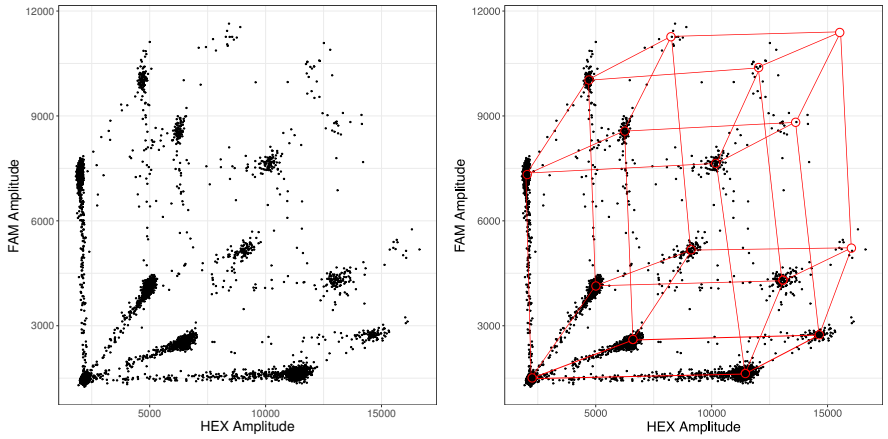
However, in practice this is not accurate due to experimental variances, sample degradation, etc. In order to accurately quantify the targets, it is not sufficient to correctly solve Equation 5.2. Each cluster label l must also be assigned correctly to the respective targets T, as presented in Equation 5.4. This defines, which genetic targets each cluster represents.

$$\sigma : L \to T, l \mapsto t \tag{5.4}$$

Identifying and assigning clusters to targets is not trivial, since the only source of information is the fluorescence footprint of the droplets. Furthermore, the precise number of clusters that are present in each dataset is unknown, due to variations in DNA concentration and amplification. The maximal number of clusters for a 4-plex ddPCR reaction is 16 (see Equation 5.1), but it could be less.

The next problem is detection and correct assignment of rain (see Section 4.2.3). Rain occurs between clusters that share at least one target (see Figure 5.1). Thus, the number of vectors connecting the clusters, which have to be considered for the detection of rain, follows from Equation 5.1 by adding the number of possible combinations for each cluster, resulting in Equation 5.5.

$$\sum_{k=0}^{t} \binom{t}{k} k = 2^{t-1} t \tag{5.5}$$



(a) Raw data (b) Vectors containing rain

Figure 5.1: Graphical representation of the formation of rain along vectors. (a) showcases an example of raw data that suffers from sample degradation, causing the formation of rain. (b) highlights the vectors connecting clusters that contain the same targets along which rain can occur.

Once all droplets are correctly assigned, copies per droplet (CPDs) for each target can be calculated according to HUGHESMAN et al. (2016), as presented in Equation 5.6 and Equation 5.7,

$$CPD_i = -\ln(1 - \frac{C_i}{C_T}) \tag{5.6}$$

where C_i is the total number of positive droplets for target i and C_T the total droplet count. The ratio for each target of interest i versus a stable reference control r then follows as

$$\ln(R_{i/r}) = \ln(\frac{CPD_i \cdot C_T}{CPD_r \cdot C_T}) = \ln(\frac{CPD_i}{CPD_r}) \tag{5.7}$$

To summarize, I define the following steps that need be addressed in this chapter:

1. Solve ρ and assign a label l to each droplet x.
2. Solve σ and assign one or multiple targets t to each cluster c.
3. Allocate the rain for each cluster c.
4. Determine the number of positive droplets for each target t and calculate the CPDs.

5.3 METHODS

In this section, I present the methods used in the *ddPCRclust* package as a solution to the problem statement. I follow the aforementioned four steps and simplify them as: clustering, cluster labeling, rain allocation, and CPDs calculation.

5.3.1 *Input data*

The input data for the analysis are one or multiple comma-separated values (CSV) files containing the raw data from ddPCR experiments. Each file represents a data frame with two dimension, one for each color channel. Each row within the data frame represents a single droplet, each column the respective intensities per color channel.

Following Chapter 1, the appropriate data abstraction can be formulated as follows: The available dataset type is tables, where each row represents an item of data and each column an categorical attribute of the dataset. The attributes are quantitative and the dataset is available in form of a static file. The required action is to derive a new dataset type (i. e. clusters), targeting the similarity of the attributes (i. e. the fluorescence intensity).

5.3.2 *Step 1: Clustering*

In manual analysis, the clustering is done by gating each individual cluster using the commercial QuantaSoft™ software (see Section 4.3). This process is tedious and time consuming and is somewhat similar to the manual gating process employed in the field of flow cytometry (SUTHERLAND et al., 1996). Thus, I compared 30 state of the art clustering methods for flow cytometry data for their respective applicability to ddPCR data (AGHAEEPOUR et al., 2013). Three algorithms proved to be capable of handling ddPCR data without extensive modifications: *flowDensity*, *SamSPECTRAL*, and *flowPeaks*. I use these algorithms to perform the initial clustering and detect all potential cluster centers.

5.3.2.1 *flowDensity*

The first approach is based on the *flowDensity* algorithm published by MALEK et al. (2015). Originally designed for gating of flow cytometry data, *flowDensity* identifies cell populations in a dataset using characteristics of the density distribution (i. e. the number, height, and width of peaks and the slope of the distribution curve). Parameters can be adjusted on a population-specific basis. I use the density function to find local peaks above a threshold, which represent the centers of clusters. Let $x = (x_1, x_2)$ be a data point in the 2-dimensional color space of this experiment. The method comprises the following steps:

1. Remove all x where $(x_1, x_2) < 0.125 \cdot max(x_1, x_2)$. The bottom 12.5 % of the data space is known to contain the negative population, i. e. the droplets without any of the targets of interest.
2. Find the highest density peaks with $max(x_1)$ and $max(x_2)$, respectively. I define these as the two outer primary clusters y and z, since the primary clusters empirically contain the majority of the non-negative events.
3. Rotate the data with $\theta = |atan(\frac{y_2 - z_2}{y_1 - z_1})|$.
4. Cut the rotated data above the two outer clusters in a staircase manner and find all density peaks (see Figure 5.2).
5. Take the previously removed data and repeat steps 2 and 4, until all clusters are found.

If too few events remain after cutting the data in step 4, the density function is rendered useless. Instead, Equation 5.3 is used to estimate the fluorescence footprint of the remaining cluster centers.

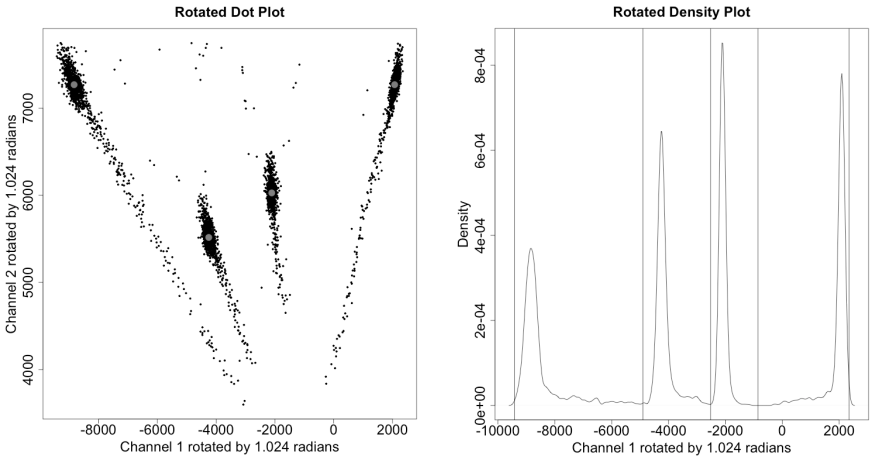


Figure 5.2: Example for the density distribution of four primary clusters, after the data has been rotated and filtered. The individual data points can be seen on the left, the corresponding local density on the right. A gray dot marks the cluster centers, identified according to the density peaks.

5.3.2.2 *SamSPECTRAL*

The second approach is built upon the clustering algorithm *SamSPECTRAL*, a version of spectral clustering adapted to flow cytometry data. It was published by Zare et al. (2010) and is available as an R package.

Since spectral clustering is computationally expensive ($\mathcal{O}(n^3)$ time and $\mathcal{O}(n^2)$ space), *SamSPECTRAL* uses density based pre-processing to reduce the number of edges in the graph. To do so, a faithful sampling algorithm builds m communities, which are then connected to a graph where the edges represent the similarity between corresponding communities. The spectrum of this graph is subsequently analyzed using classical spectral clustering to find the clusters. Finally, the clusters are combined based on their similarity in the community graph and a cluster number for each event in the original data is returned. I use this implementation of spectral clustering and choose m encompassing 5 % of the data, which has empirically proven to be a good compromise between accuracy and speed. However, users can choose a different value if necessary.

5.3.2.3 *flowPeaks*

The third approach uses the *flowPeaks*Peaks package for R, developed by Ge and Sealfon (2012). The *flowPeaks* algorithm first uses a two step k-means clustering with a large k, in order to partition the dataset into many compact clusters. The result is then used to generate a smoothed density function. All local peaks are exhaustively found by exploring the density function and the clusters are merged according to their local peaks.

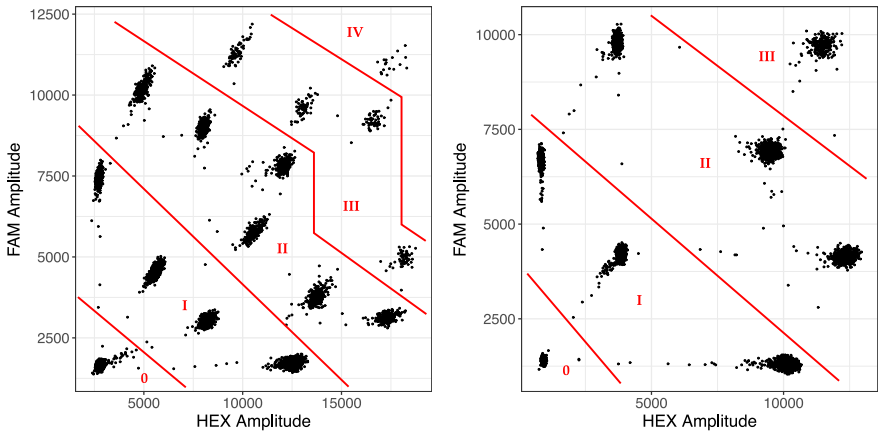
5.3.3 *Step 2: Cluster labeling*

After the clustering is done, targets need to be assigned to each cluster. Clusters that contain no target are defined as the empty population. Clusters that contain only a single target are defined as primary clusters. Subsequently, clusters that contain two targets are defined as secondary clusters, clusters that contain three targets are defined as tertiary clusters, and the cluster that contains all four targets is defined as quaternary cluster, as presented in Figure 5.3.

Because the *flowDensity* approach already employs rotation to find the density peaks, labeling of the clusters can be performed at each step of the clustering method (see Figure 5.2).

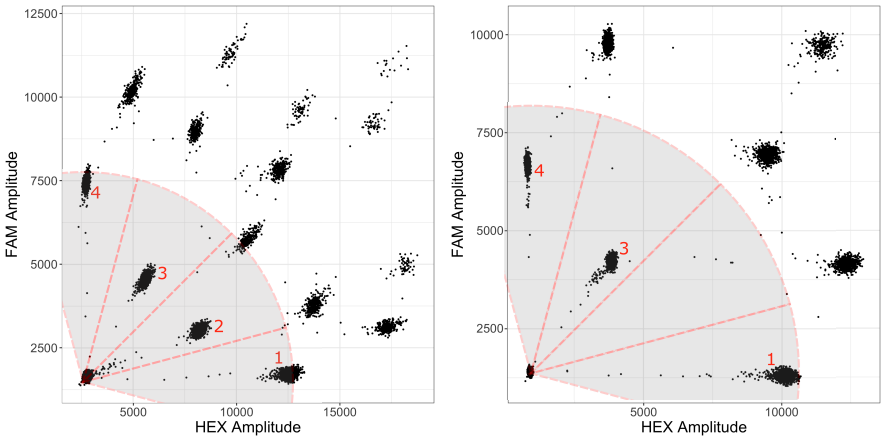
For the *flowPeaks* and *SamSPECTRAL* approaches, this needs to be resolved based on the relative position of each cluster. Due to the particular layout of the data, the angle between the population of empty droplets and the respective first order clusters seems the straightforward and most efficient way to label the first order clusters (see Figure 5.4).

I can estimate for the position of every cluster once the location of the primary clusters is known based on Equation 5.3. I use this estimate to create a distance matrix m, containing the distances between the estimated cluster positions d and all cluster centers c found by the density of the peaks. The optimal assignment for each cluster can then be calculated by solving the so called Linear Sum Assignment Problem using the Hungarian Method (Papadimitriou and Steiglitz, 1982).



(a) 4-plex reaction (b) 3-plex reaction

Figure 5.3: Graphical description of the different cluster categories: (0) empty population, (I) primary clusters, (II) secondary clusters. (III) tertiary clusters, (IV) quaternary cluster. A 4-plex reaction (a) contains all categories, a 3-plex reaction (b) only contains empty to tertiary clusters.



(a) 4-plex reaction (b) 3-plex reaction

Figure 5.4: The angles between the droplets on the bottom left, which retain no target, and the first order clusters are highlighted. Based on the relative position of the respective cluster centers, I determine which targets these represent. In case of genomic deletions or purposely missing clusters, it is possible to determine which cluster is missing. In case (b), a genomic deletion of target 2 has occurred.

5.3.4 *Step 3: Rain allocation*

As presented in Section 4.3, ddPCR experiments with DNA obtained from clinical FFPE samples involve rain, which can contain up to half of the droplets intrinsically belonging to the higher order cluster in case of profound sample degradation. Hence, accurate allocation of rain is a crucial part of the *ddPCRclust* algorithm. To do so, we have to find the minimal distance between each droplet and each cluster, as well as between each droplet and the respective vectors connecting the clusters (see Figure 5.1). The naive solution is an all-vs-all comparison:

1. Go through the whole clustering result and for each row, calculate the distance to all cluster centers and all vectors.

2. If the nearest element is a vector, compare the two minimal distances for the data point. If their fraction is $\geqslant 95\,\%$, the row will be not be assigned to any cluster, but removed from the final result.

3. Assigned the point to the nearest cluster or the cluster, to which the nearest vector belongs.

However, this function has significant impact on the runtime of the algorithm. The number of comparisons necessary in this step can be estimated by combining Equation 5.1 and Equation 5.5, multiplied with the size of the dataset (n), which yields $\mathcal{O}(2^{t-1}(t+2)n)$. For an

average experiment with $t = 4$ and $n = 15\,000$, this leads to $720\,000$ operations per file. However, this number can be reduced by preprocessing the data. Filtering out points that are obviously not rain, can greatly demagnify n, speeding up the algorithm significantly in the process. The obvious choice are points that are sufficiently close to the cluster centers.

To estimate the distance of a point to a cluster center, I use the Mahalanobis distance (MAHALANOBIS, 1936). The reason for using the Mahalanobis distance instead of the standard deviation is that it does not require the clusters to be spherical. Furthermore, only taking clusters and vectors in the vicinity of the data point into account will lower the number of operations even further. The whole function is comprised of the following steps:

1. For each cluster center c, calculate the Mahalanobis distance d_M to each point based on the covariance matrix of the dataset.
2. Remove all points where $d_M < 0.2$. Those points are around the respective cluster centers and hence do not have to be considered as rain.
3. For each cluster center c, remove all points that are not in between c and the respective higher order clusters.
4. For all remaining points, calculate the minimal distance in an all-vs-all comparison as described earlier.

The intermediary result are three arrays of unique identifiers, which represent the cluster membership for each row of the data frame. Each array is the result from one of the three independent clustering approaches. Next, these results need to be combined.

5.3.5 *Step 4: CPDs calculation*

Until this point, all three approaches were computed independently. To compute the final result, I create a *cluster ensemble*. A cluster ensemble, sometimes also referred to as consensus clustering or aggregation of clustering, is a collection of individual solutions to a given clustering problem. For *ddPCRclust*, this provides a number of benefits. If one approach performs poorly, it will be compensated by the other two approaches. Furthermore, it provides means to display a measure of confidence, by calculating the agreement of the three approaches. If the agreement is high, the final result is likely to be correct.

The cluster ensemble is calculated using the *clue* package for R (HORNIK, 2005). The results of the previous clusterings are first converted into partitions, before the mediod of the cluster ensemble is computed, i. e. the element of the ensemble minimizing the sum of dissimilarities to all other elements. As a measure of confidence, the

agreement of the cluster ensemble is calculated using the adjusted Rand index (Hubert and Arabie, 1985; see Equation 6.2).

Once all droplets are assigned, the CPDs for each target can be calculated according to Equation 5.6. In order to compare individual wells (or files) with each other, a constant reference control is required. This target should be a genetic region that is usually not affected by any variations and present in every file. If the name of this marker is provided, all CPDs will be normalized against that control.

6 DDPCRCLUST RESULTS

In the previous chapter, I presented *ddPCRclust*, and R package for automated analysis of multiplexed ddPCR samples. In this chapter, I present clustering results on real datasets, comparing the *ddPCRclust* algorithm to manual annotation. I also discuss these results and give an outlook on possible applications. All automatic results have been computes an a MacBook Air with Intel(R) Core(TM) i7-4650U CPU @ 1.70GHz and 8 GB RAM. Manual droplet counts have been obtained by experts using the commercial QuantaSoft^{TM} software.

6.1 RESULTS

I compared four datasets (D1–D4) comprised of a total of 360 individual reactions. However, the manual annotation method described in Section 4.3 only provides the number of positive droplets per target and does not provide a cluster label for each event. Therefore, I have to use a custom metric. For each reaction r, I calculate the difference d_r according to Equation 6.1,

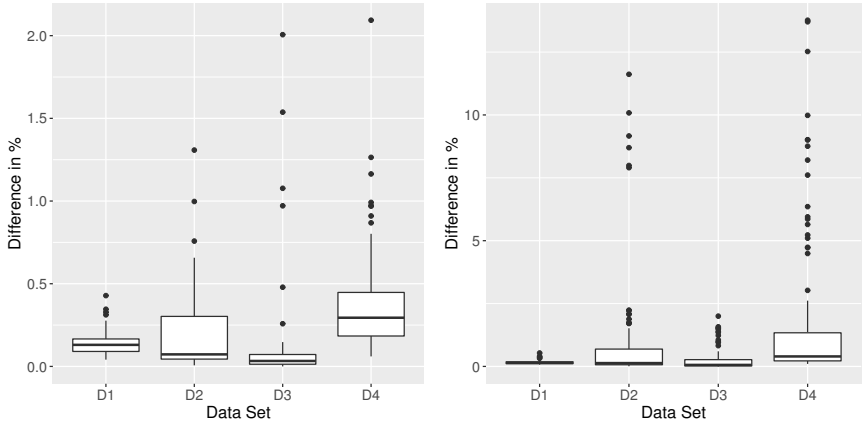
$$d_r = \frac{\sum_{i=1}^{t} |auto_i - man_i|}{total_r} \cdot 100 \tag{6.1}$$

where t is the number of targets in this reaction, $auto_i$ the number of positive droplets for target i according to the *ddPCRclust* algorithm, man_i the number of positive droplets for target i according to manual annotation and $total_r$ the total number of droplets in this reaction. Based on this, I calculated the percentage of differently assigned droplets in each dataset. I compare both the full algorithm with all three clustering approaches, as well as the fast mode, which only uses the flowDensity based approach (see Chapter 5). The results can be seen in Figure 6.1. The run time of the algorithm has been computed as well for each dataset and is shown in Table 6.1.

Table 6.1: Run time of *ddPCRclust* for selected datasets.

	D1	D2	D3	D4
Number of reactions	72	96	96	96
Fast mode run time	67 s	17 s	13 s	18 s
Full mode run time	333 s	309 s	360 s	144 s

Along with the R package, a set of eight representative files of ddPCR data is provided. For these eight files, each row (i. e. each



(a) Automatic clustering vs manual (b) Fast mode vs manual

Figure 6.1: The difference between automatic and manual analysis for selected datasets. In (a) the results for the full algorithm are shown, while (b) presents the results for the fast mode. Dataset D1 comprises 72 reactions, the others comprise 96 reactions each.

droplet) has been assigned manually to its respective cluster by experts. Thus, it is possible to compare the clustering results of *ddPCRclust* to manual analysis using the adjusted Rand index (Hubert and Arabie, 1985). The adjusted Rand index (ARI) is the corrected-for-chance version of the Rand index, a measure of similarity between two data clusterings, and is defined as follows.

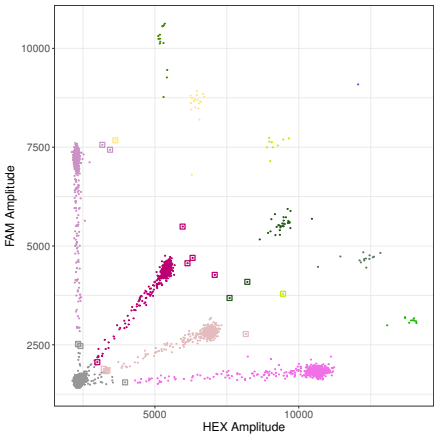
Given a set S of n elements, and two clusterings of these points, namely $X = \{X_1, X_2, \ldots, X_r\}$ and $Y = \{Y_1, Y_2, \ldots, Y_s\}$, the overlap between X and Y can be summarized in a contingency table $\left[n_{ij}\right]$ where each entry n_{ij} denotes the number of objects in common between X_i and Y_j : $n_{ij} = |X_i \cap Y_j|$.

$$\left[n_{ij}\right] = \begin{array}{c|cccc|c} {}_X\backslash^Y & Y_1 & Y_2 & \ldots & Y_s & \text{Sums} \\ \hline X_1 & n_{11} & n_{12} & \ldots & n_{1s} & a_1 \\ X_2 & n_{21} & n_{22} & \ldots & n_{2s} & a_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ X_r & n_{r1} & n_{r2} & \ldots & n_{rs} & a_r \\ \hline \text{Sums} & b_1 & b_2 & \ldots & b_s & \end{array}$$

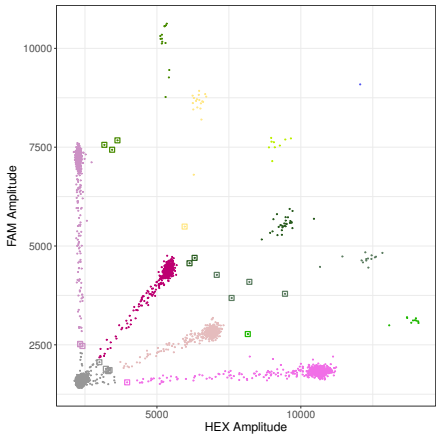
$$\text{ARI} = \frac{\sum_{ij}\binom{n_{ij}}{2} - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}] - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{n}{2}} \tag{6.2}$$

The result for those eight reactions is presented in Table 6.2. Furthermore, a visual comparison between the results is presented in Figure 6.2, where the differences between the clusterings are highlighted.

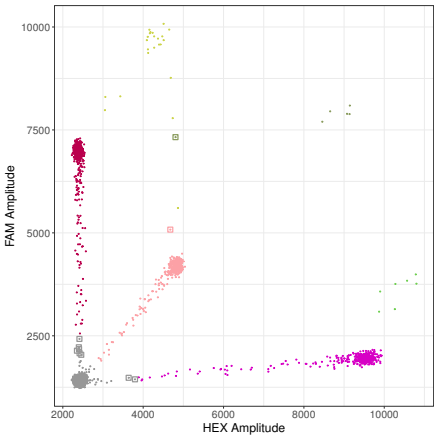
Due to pending patents, it is not possible to present real biomarkers and evaluate *ddPCRclust* in terms of its performance to actually detect relevant changes in biomarkers that are from cancerous samples.



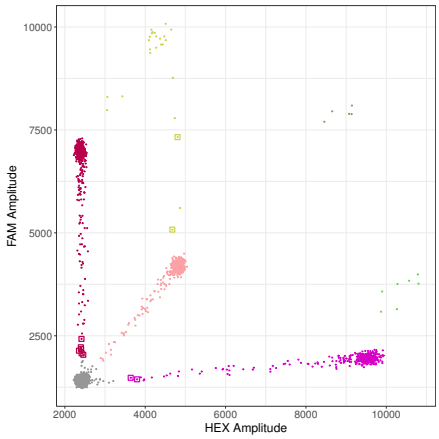
(a) Example file 1 (automatic)



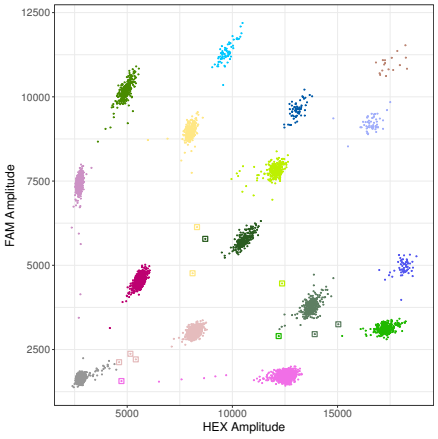
(b) Example file 1 (manual)



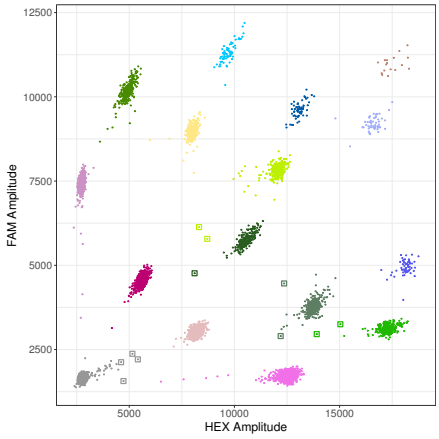
(c) Example file 2 (automatic)



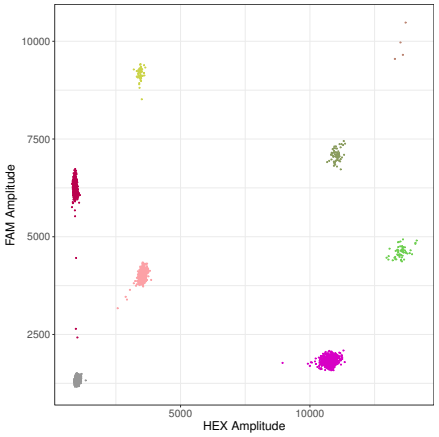
(d) Example file 2 (manual)



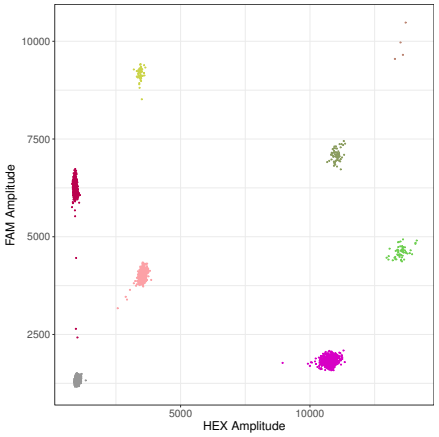
(e) Example file 3 (automatic)



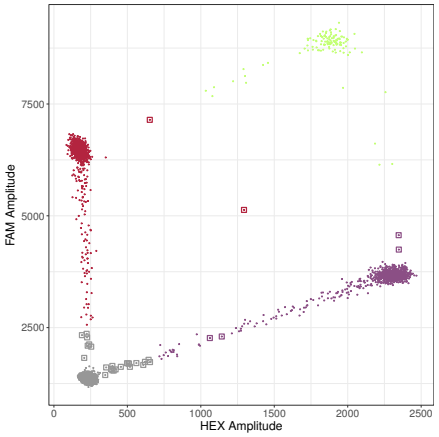
(f) Example file 3 (manual)



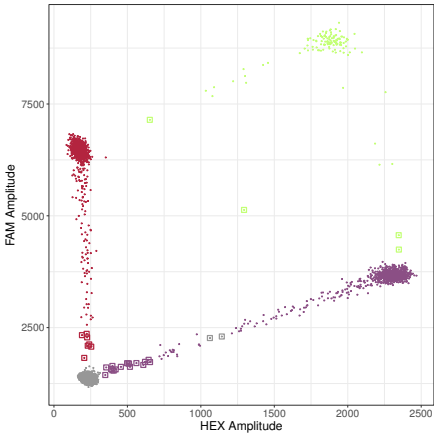
(g) Example file 4 (automatic)



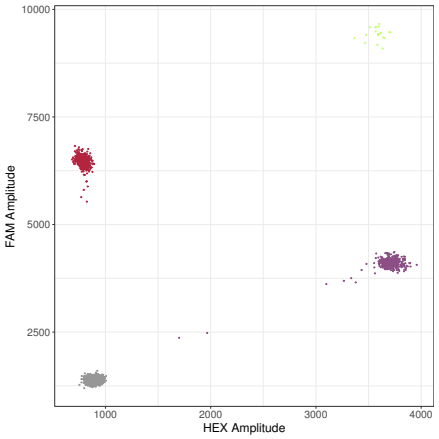
(h) Example file 4 (manual)



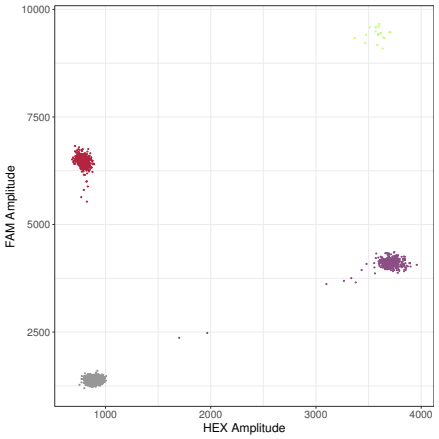
(i) Example file 5 (automatic)



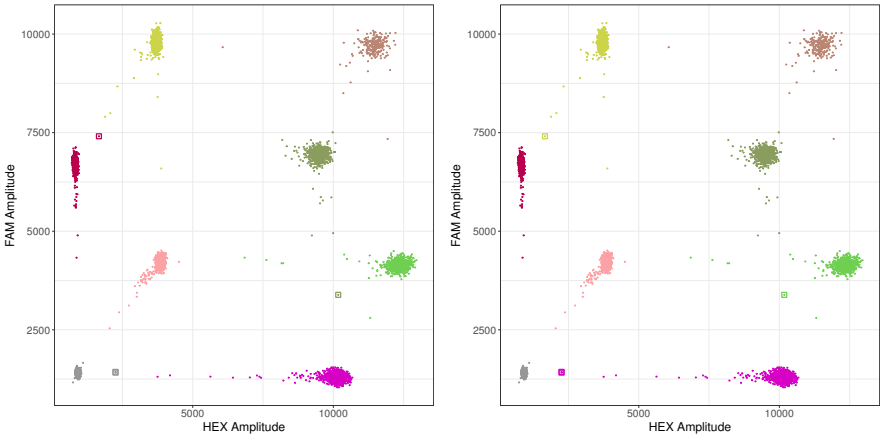
(j) Example file 5 (manual)



(k) Example file 6 (automatic)

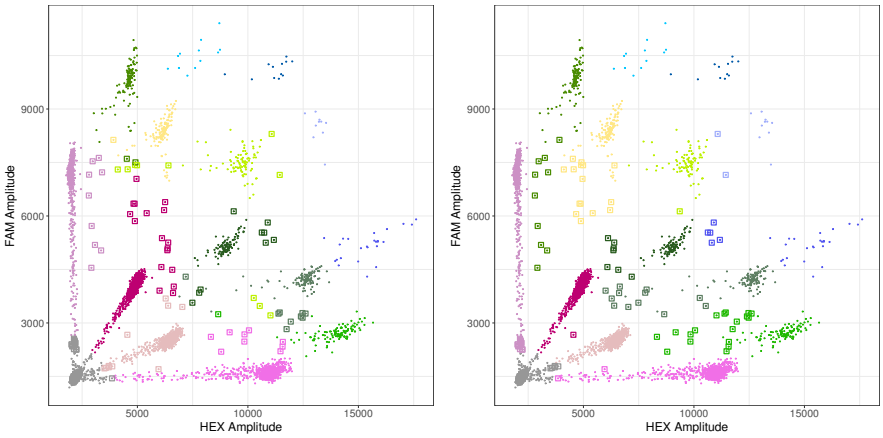


(l) Example file 6 (manual)



(m) Example file 7 (automatic)

(n) Example file 7 (manual)



(o) Example file 8 (automatic)

(p) Example file 8 (manual)

Figure 6.2: Comparison between automatic and manual annotation for the eight example files included in the package. Each point represents one droplet and its respective intensities per fluorescence channel. The color represents cluster membership (see Figure 7.8). Differences are highlighted with a rectangle.

Table 6.2: Run time and accuracy compared to manual annotation by experts for eight exemplary reactions provided alongside the R package. Each entry comprises the mean and the standard deviation, the latter being in brackets.

Total number of droplets	Adjusted Rand index	Run time in seconds
14590 (1295)	0.997 (0.003)	7.18 (1.98)

6.2 DISCUSSION

While the advantages of digital PCR in terms of sensitivity and accuracy have already been established, the technology has long been held back by its low throughput compared to other techniques. The advancements of using thousands of nanoliter droplets instead of physical wells paired with new protocols for multiplexed ddPCR reaction will provide a massive boost to the field of digital PCR. These new types of data require new computational methods to be devised in order to avoid a bottleneck on the analysis end of the technology. Automatic analysis of non-orthogonal reactions using the commercial QuantaSoft™ Software is impossible and manual analysis takes many hours to complete, while suffering the usual disadvantages of subjectivity and non-reproducibility.

I developed *ddPCRclust*, an R package which can automatically calculate CPDs for multiplexed ddPCR reactions with up to four targets in a non-orthogonal layout. I showed that the results of *ddPCRclust* are on a par with manual annotation by experts, while the computation only takes a few minutes per 96-well experiment. Three independent clustering approaches provide robustness, which is especially important in a medical context. This is an important first step in supporting this new technology, and I am certain once it becomes more widespread, faster and more efficient methods will be developed.

As presented in Figure 6.1, the mean difference over all 360 reaction between *ddPCRclust* and manual analysis is ~0.21 %, being as low as ~0.10 % for D3, while performing worst for D4 with a mean difference of ~0.37 %. It becomes evident that the accuracy of the algorithm depends on the quality of the data. Profound sample degradation, mostly happening in formalin-fixed paraffin-embedded clinical samples, causes the formation of rain. Furthermore, it has a negative impact on the compactness of the clusters. This fact becomes especially clear for the fast mode, where the cleaner datasets D1 and D3 provide much better results than D2 and D4. The run time underlines the fact that D4 suffered from a low amplification efficiency, being more than twice as fast as the other datasets.

The comparison of the eight example files presented in Table 6.2 and Figure 6.2 highlights the strengths and weaknesses of the *ddPCRclust* algorithm. In the cases of files 3 and 5, the automatic solution proves to be superior to the manual annotation. It has been established that in the case of rain, 20 % of droplets shall be accounted for the lower cluster and 80 % for the higher cluster. This is difficult to estimate precisely when annotating the data manually, where the uneven scaling of the axes provides an additional hindrance. The automatic solution has an advantage here, because it can calculate the distance precisely for each droplet. Thus, the droplets close to the negative cluster in the bottom left are assigned correctly in the automatic

case, but wrong in the manual case. Example file 8 however proves that intensive rain can be difficult for both automatic and manual annotation. There are many differences visible between the primary and secondary clusters. In this case, the manual annotation is probably closer to the truth and the option to manually correct some of the automatic results would be beneficial. Moreover, a manual correction of critical wells by experts would still be faster than annotating everything by hand and provide the same accuracy. That's part of the reason why I developed a the GUI *ddPCRvis* based on the Shiny technology, which will be presented in the next chapter.

6.3 OUTLOOK

Detecting cancer in its earliest stages can drastically increase chances for a successful treatment and recovery. Today, technology such as ddPCR is providing means for detecting even subtle changes to the DNA. Discovering CNAs is part of promising research in these directions. Taking a sample from equivocal tissue can lead to detection of somatic mutations, even before an actual tumor forms.

Furthermore, it has been shown that neoplastic diseases cause an increase of cell-free nucleic acids circulating in the blood of patients (Schwarzenbach, Hoon, and Pantel, 2011). This is believed to be caused by the apoptosis and necrosis of cancer cells in the tumor due to acquisition of lethal mutations and by the immune response of the patient. It has been shown for different cancers (e. g. breast cancer (Beaver et al., 2014), lung cancer (Oxnard et al., 2014), oral cancer (Hughesman et al., 2016)) that is possible to detect cell-free tumor DNA in the blood with the help of ddPCR. If more cancer related biomarkers are discovered and the technology for detecting CNAs and other mutation is further refined, it could be possible to detect many cancers in their earliest stage simply by analyzing a blood sample. This could revolutionize cancer check-up procedures, since taking a patient's blood sample on a regular bases would suffice.

7

THE VISUAL INTERFACE DDPCRVIS

I presented in Chapter 2 that an interactive visual interface is crucial for the users to get a mental model of their data and make the tool accessible. I therefore developed a GUI, which provides access to the functionality of the *ddPCRclust* package directly through a web browser. It is build upon Shiny, a web application framework for R. I call the GUI *ddPCRvis* and it is available under `https://bibiserv.cebitec.uni-bielefeld.de/ddPCRvis/`.

7.1 IMPLEMENTATION

The *shiny* package for R enables developers to create a web application using the R programming language (Chang et al., 2017). The required Hypertext Markup Language (HTML), Cascading Style Sheets (CSS) and JavaScript elements are generated directly from R code. Shiny applications have two components: a user-interface definition and a server script. The user interface is defined in a source file named `ui.R`, the server side of the application is defined in a source file named `server.R`.

7.1.1 *Structure of a Shiny application*

As presented in Chapter 2, interactivity is a crucial aspect of any modern, computer based visualization. In the context of a Shiny application, interactivity means that the input values can change at any time, and the output values need to be updated immediately to reflect those changes. To achieve this, Shiny applications use a concept called *reactive programming*, where inputs and outputs are connected together "live" and changes are propagated immediately.

7.1.2 *Reactive programming*

Reactive programming is a coding style, which revolves around three important aspects: reactive sources, reactive conductors, and reactive endpoints (Figure 7.1). Reactive sources are values that change over time, or in response to the user. Reactive conductors are expressions that access reactive sources and execute other reactive conductors. Reactive endpoints can access reactive sources and reactive conductors, but they don't return a value.

From an implementation point of view, these aspects are represented by reactive values, reactive expressions, and observers. The im-

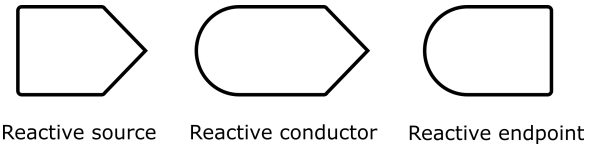


Figure 7.1: Objects in reactive programming: reactive sources, which are implementes by reactive values; reactive conductors, which are implementes by reactive expression; and reactive endpoints, which are implemented by observers.

portant thing about reactive expressions is that they are self-updating features that keep track of changes. They automatically keep track of what reactive values they read and what reactive expressions they invoked. If a dependency becomes out of date, they know that their own return value has also become out of date. Because of this dependency tracking, changing a reactive value will automatically instruct all reactive expressions and observers that directly or indirectly depended on that value to re-execute.

In a Shiny application, the source typically is user input through a browser interface. For example, when the user selects an item, types input, or clicks on a button, these actions will set values that are reactive sources. A reactive endpoint is usually something that appears in the user's browser window, such as a plot or a table of values. This relationship can be formulated as a graph. A simple example for this is given in Figure 7.2.

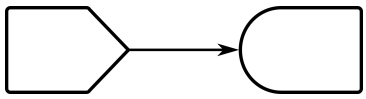


Figure 7.2: Simple example for a graph of the reactive structure. One reactive source (value) is connected with one reactive endpoint (observer).

It is also possible to put reactive components in between the sources and endpoints. These components are called reactive conductors. A conductor can both be a dependent and have dependents. In other words, it can be both a parent and child in a graph of the reactive structure. Sources can only be parents (they can have dependents), and endpoints can only be children (they can be dependents) in the reactive graph. Reactive conductors can be useful for encapsulating slow or computationally expensive operations, for instance calculating the nth element of the Fibonacci series.

In the case of *ddPCRvis*, I use conductors for example to parallelize the computation of the *ddPCRclust* algorithm. Each clustering approach is launched multiple times, depending on the number of CPU cores, such that multiple files are being evaluated at once (Figure 7.3). Another example would be the editing of individual results, where the user interacts with one plot and a second plot is updated

live according to these changes (see Section 7.2.3). A conductor connects these two endpoints and the underlying data items (*Shiny - Reactivity - An overview*).

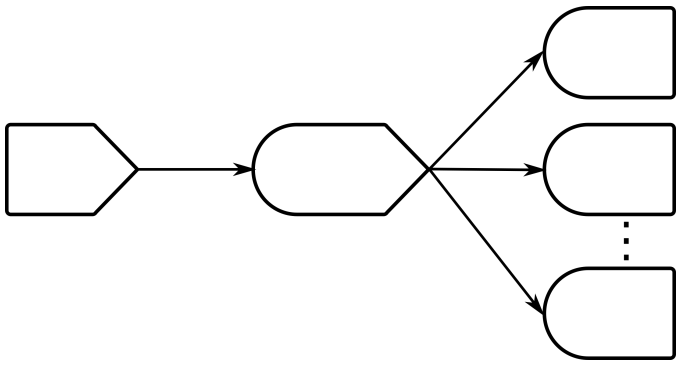


Figure 7.3: A reactive conductor can link reactive sources and endpoints. In *ddPCRvis*, I use it to run each clustering approach from the *ddPCRclust* package multiple times in parallel.

7.2 THE WEB INTERFACE

The website is divided into six pages: Upload Files, Clustering, Edit Clustering, Counts, CPDs, and Results Figure 7.4. Each page can be accessed using the navigation bar on the top. I will present each view and explain the design principle based on the questions presented in Chapter 1:

- What is to be visualized?
- Why visualize it?
- How to visualize it?

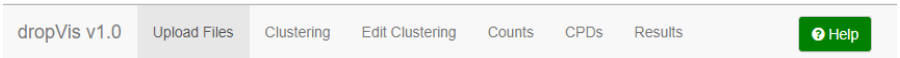
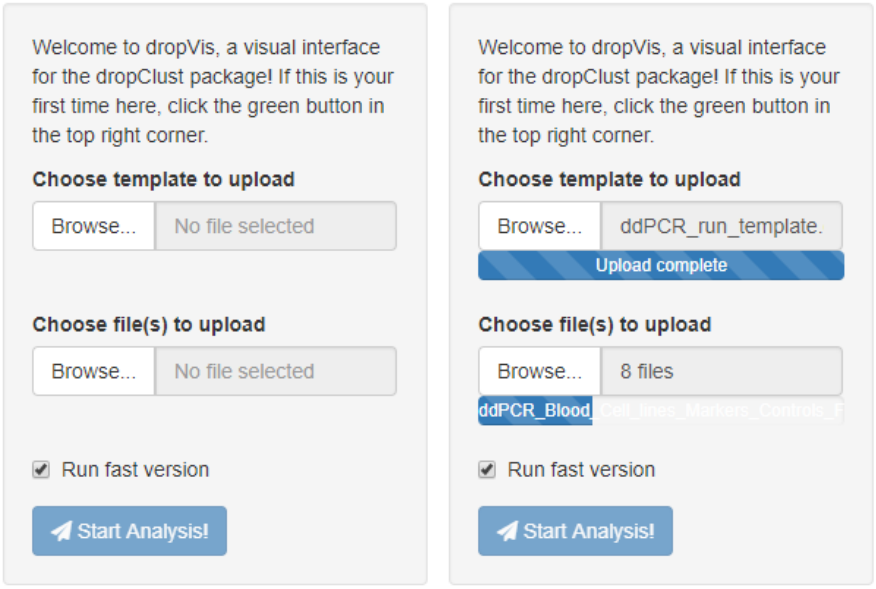


Figure 7.4: Navigation bar of the *ddPCRvis* application. The website is divided into six views: Upload Files, Clustering, Edit Clustering, Counts, CPDs, and Results. The green button on the right starts the dynamic help system for the current view.

7.2.1 *Upload files*

Uploading the raw data is the first step, if any kind of analysis shall be performed on it. Hence, the first view the users see when they launch *ddPCRvis* is the *Upload Files* view. A control panel on the left side is present in every view and gives the users access to all the functions available to them on the current page. Here, it enables uploading

both the raw data and a template file, which specifies the setup of the ddPCR reactions for this experiment (Figure 7.5a). Once some files have been selected, a progress bar underneath the input field gives the users direct feedback on his actions (Figure 7.5b).



(a) Before file upload (b) During file upload

Figure 7.5: Example for a control panel on *ddPCRvis*. Here, the users can upload their template and raw data. Progress bars give direct feedback.

Table 7.1: Example for a run template for both *ddPCRvis* or *ddPCRclust*.

```
> Experiment_name, channel1=HEX, channel2=FAM, annotations(date, etc.)
```

Well	Sample type	No of markers	Marker 1	Marker 2	Marker 3	Marker 4
B01	Blood	4	a	b	c	d
G01	FFPE	4	a	b	c	d
F02	Blood	3	a		c	d
D03	FFPE	3	a		c	d
A04	FFPE	4	a	b	c	d
G07	Cell line	3	a		c	d
G08	Cell line	3	a		c	d
E09	FFPE	2			c	d

The available raw data for this project are tables, where each row represents an item of data and each column an categorical attribute of the dataset (see Section 1.2.1). The same is true for the template, which is to be uploaded or designed directly within *ddPCRvis*, in order to specify the details of this experiments. Since one experiment most likely consists of many different files, naming them appropri-

ately is important in order to keep things organized. I chose to use a unique identifier in each filename of the form:

```
"^[[:upper:]][[:digit:]][[:digit:]]$"
```

(A01, A02, A03, B01, B02, ...), which is usually included automatically by the ddPCR device. These identifiers are then referred to in the template, which should follow the layout presented in Table 7.1. Thus, the visualization must give the users an overview over this setup (see Section 1.2.2), which is realized using a simple, interactive table view (Figure 7.6). The users can spot and edit any mistakes in the setup and select, which files are about to be analyzed. Finally, the analysis can then be started, by clicking the respective button as seen in Figure 7.5.

Well	Sample.type	X..of.markers	Marker.1	Marker.2	Marker.3	Marker.4	Analyze
B01	Blood	4	a	b	c	d	☑
G01	FFPE	4	a	b	c	d	☑
F02	Blood	3	a		c	d	☑
D03	FFPE	3	a		c	d	☑
A04	FFPE	4	a	b	c	d	☑
G07	Cell line	3	a		c	d	☑
G08	Cell line	3	a		c	d	☐
E09	FFPE	2			c	d	☐

Figure 7.6: Interactive table view of a template in *ddPCRvis*. Cells can be edited and files selected or deselected for analysis.

7.2.2 *Clustering*

After the clustering algorithm has finished, the users are automatically redirected to the next page *Clustering*. The main view on this page is comprised of three columns. On the left side a gray on white image of the raw data is be displayed. Next to it, a colored image is shown, where each color represents cluster membership for the respective droplet. On the right side, a percentage between 0 and 100 is shown. This represents the agreement between the different underlying clustering algorithms and serves as a measure of confidence (see Figure 7.7).

Following the aforementioned design principles, the data types to visualize here are clusters. The users want to get an overview over the results and identify outliers or poor clustering results. I encode the data in a scatter plot and highlight cluster membership using color. To visualize 16 clusters, I designed a custom color palette, which is presented in Figure 7.8. Green and yellow have been combined to one category, due to yellow being poorly visible on white background. Furthermore, shades of green are known to have the widest range of

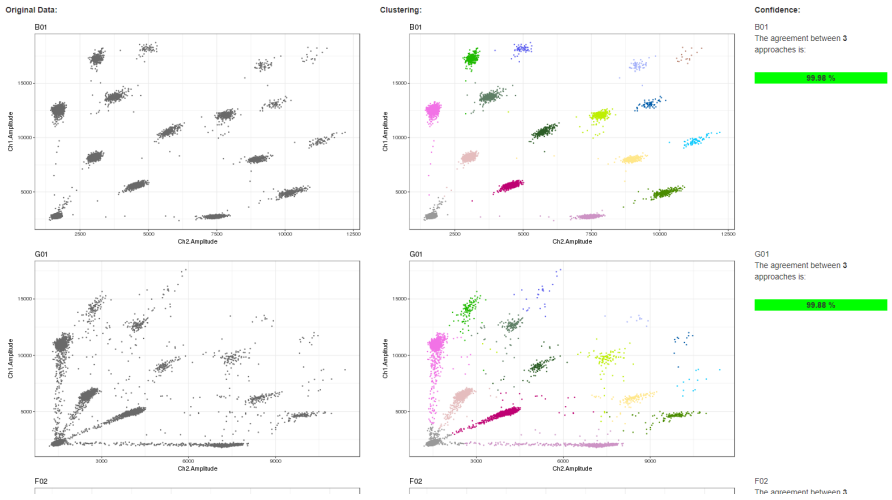


Figure 7.7: Main view of the clustering results. On the left side a gray on white image of the raw data is be displayed. Next to it, a colored image is shown, where each color represents cluster membership for the respective droplet. On the right side, a percentage between 0 and 100 is shown. This represents the agreement between the different underlying clustering algorithms and serves as a measure of confidence (see Section 5.3.5).

perception for the human eye (Ware, 2010). Thus, I use the combination of the green and yellow spectrum for the category with the highest number of clusters: the secondary clusters. According to the opponent process theory (see Section 1.2.3), I chose shades of red and shades of blue for four primary and tertiary clusters, respectively. The empty population is represented by a gray color and the quaternary cluster has red-brown shade, which is again opposing the blue shades of the neighboring tertiary clusters.

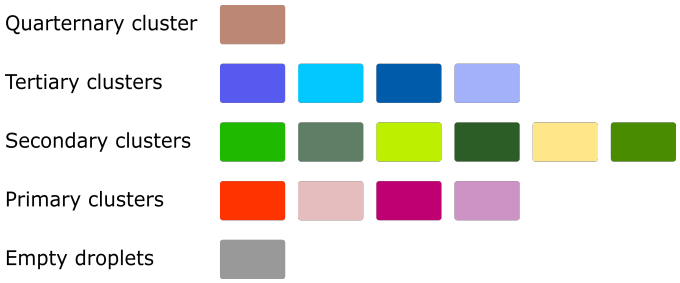


Figure 7.8: The color pallete for visualizing the clustering results. The clusters can be divided in 5 sub-categories: Empty population, primary clusters, secondary clusters, tertiary clusters, and quaternary cluster. Each sub-category is represented by shades of a color family.

The agreement between the algorithms is also emphasized with color, using a common palette from green over yellow and orange to red. The encoding is defined as follows:

- $> 98\,\%$ agreement $\rightarrow$ green
- $> 95\,\%$ agreement $\rightarrow$ yellow
- $> 90\,\%$ agreement $\rightarrow$ orange
- else $\rightarrow$ red

7.2.3 *Edit clustering*

As presented in Section 5.2, a crucial aspect of the algorithm is correct assignment of droplets to their respective clusters. Means for detecting errors have been presented in Figure 7.7, but the users also need to be able to correct them. I therefore implemented an interface, where users can interact with the visualization and manually reassign any number of droplets to any of the clusters. The data types and task abstraction are the same as in Section 7.2.2, but I add another interface, with which the users can interact and select droplets that they want to reassign.

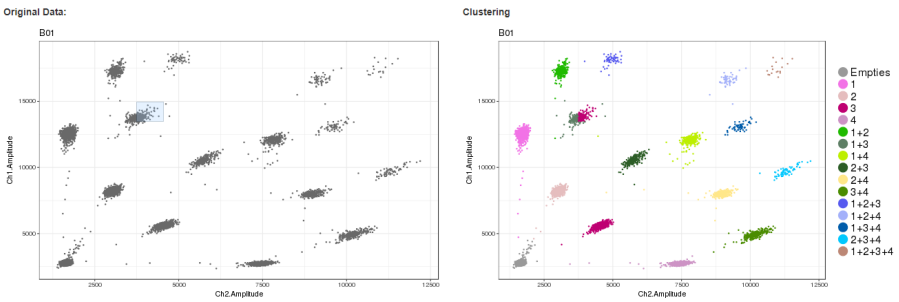
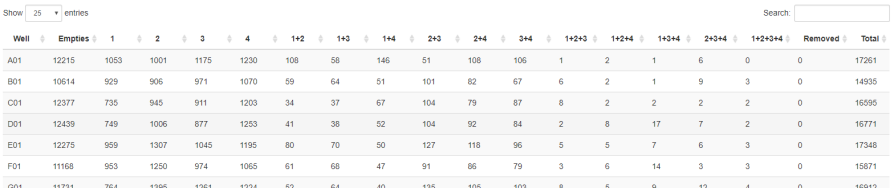


Figure 7.9: Main view of the Edit Clustering page in *ddPCRvis*. On the left side, a gray on white representation of the raw data is displayed. Users can select any area by clicking with your mouse and holding the button to draw a rectangle. These droplets will be reassigned to the cluster you selected in the menu on the left. The colored plot to the right is updated live, so users can check if their changes are correct. In this example, part of cluster "1+3" has been manually assigned to cluster "3".

7.2.4 *Counts*

The next page presents the raw counts for this experiments. The data type in this visualization is again a table, where each row represents one file that has been uploaded and each column one cluster or the total number of droplets, respectively. The table view was chosen as an appropriate presentation, including means to search and sort it (see Figure 7.10).

The control panel on the left side enables the users to download the counts as CSV files and select a stable reference control, which is


Show 25 entries Search:

Well	Empties	1	2	3	4	1+2	1+3	1+4	2+3	2+4	3+4	1+2+3	1+2+4	1+3+4	2+3+4	1+2+3+4	Removed	Total
A01	12215	1053	1001	1175	1230	108	58	146	51	108	106	1	2	1	6	0	0	17261
B01	10614	929	906	971	1070	59	64	51	101	82	67	6	2	1	9	3	0	14935
C01	12377	735	945	911	1203	34	37	67	104	79	87	8	2	2	2	2	0	16595
D01	12439	749	1006	877	1253	41	38	52	104	92	84	2	8	17	7	2	0	16771
E01	12275	959	1307	1045	1195	80	70	50	127	118	96	5	5	7	6	3	0	17348
F01	11168	953	1250	974	1065	61	68	47	91	86	79	3	6	14	3	3	0	15871
G01	11731	764	1265	1251	1224	52	64	40	135	105	109	8	5	9	12	4	0	16842

Figure 7.10: Main view of the Counts page in *ddPCRvis*. Each row represents one file that has been uploaded and each column one cluster or the total number of droplets, respectively. The table can be searched and sorted according to each column. Users can select how many rows should be displayed at once.

used to normalize the data (see Section 5.3.5). A click on "Calculate CPDs" brings users to the next page.

7.2.5 *CPDs*

This page is similar to the page *Counts*, except now each row represents one genetic target (or marker), as presented in Figure 7.11.


Show 25 entries Search:

Well	Sample name	Marker	droplet count	CPD
A01	Sample	C1	1369	0.09521556
A01	Sample	C2	1277	0.08856428
A01	Sample	C3	1398	0.09732014
A01	Sample	CC	1599	0.11201371
B01	Sample	C1	1115	0.09660347
B01	Sample	C2	1168	0.10138741
B01	Sample	C3	1222	0.10628059

Figure 7.11: Main view of the CPDs page in *ddPCRvis*. Each row represents one target (or marker) that has been specified in the template. The columns represent the name of the file, the sample name, the marker name, raw droplet count, and CPDs. The table can be searched and sorted according to each column. Users can select how many rows should be displayed at once.

7.2.6 *Result*

The *Result* page offers two different visualizations depending on the task at hand. Initially, CPDs for each target are displayed as a box-and-whisker plot, a graphical method of displaying variation in a dataset (McGill, Tukey, and Larsen, 1978; see Section 9.3 for details). Each box represents the CPDs for one target. The size of the box represent the lower and upper quartile, and the whiskers represent the extremes. To enhance visibility, the individual boxes are also separated by color. This gives users a first overview over the variability in their data.

Based on this, they can select a number of constant controls using the navigation bar to the left. Constant controls are genes, which have been selected as targets because they are known to show little

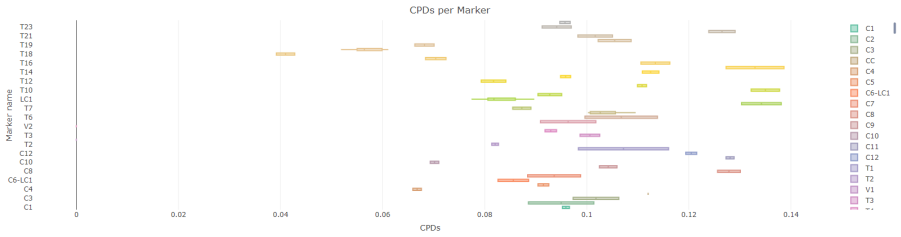


Figure 7.12: Main view of the results page displaying CPDs as a box-and-whisker plot. Each box represents the CPDs for one target.

variety even under increased genomic instability during a tumor (see Chapter 4). Once a set of controls has been selected, the visualization changes to a bar plot, because the new task abstraction is to compare targets with the selected controls. The bars represent the mean difference in expression for each target compared to the mean of the selected controls.

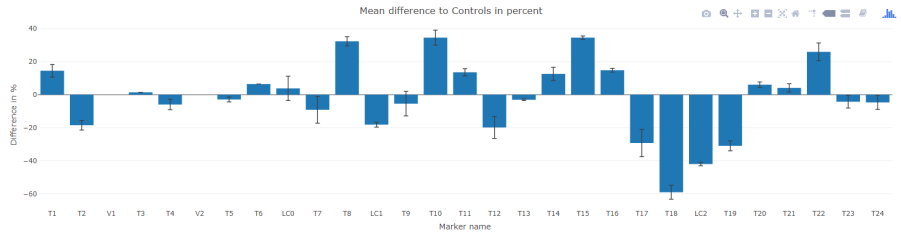


Figure 7.13: Main view of the results page displaying the difference of targets and selected controls. Each bar represents the difference in percent. If multiple replicates were provided, error bars show the variance.

7.2.7 *Dynamic help system*

Giving users feedback on their current action is an important aspect of making an intuitive visualization and GUI. However, it is always possible that a user is lost along the way, so an additional help system will provide extra support. The JavaScript library *introJS* offers means for a step-by-step guide by highlighting elements on a web page. I use this library to draw boxes around important aspects of the current view. The help text and the respective element of the GUI, where the text is referring to, are displayed in normal color, while graying out everything else. This way, emphasis is put on the the two elements and it is easy for the user to grasp the connection. Furthermore, this is combined with an annotation layer and a navigation system to click through the different steps, which can be performed on this page.

7.3 DISCUSSION

A well designed GUI can greatly improve the reach of a tool and the performance of its users (SHNEIDERMAN, 2010). I presented *ddPCRvis*, a web based GUI for the *ddPCRclust* package built on R Shiny.

Each step of the algorithm requires its own visualization, with different visual idioms. I defined the tasks for each visualization based on the questions presented in Chapter 1. Direct feedback to any actions enhances user experience. The clustering results are visualized using a custom color palette, which has been optimized for the data at hand. Interactivity is used when modifying the clustering results, giving the user access to the previous steps of the algorithm and modify the results where necessary (see Figure 2.1).

ddPCRvis does not only provides easy access to the algorithm, but it also enables the user to check the results and manually correct them if necessary. Furthermore, in a medical context supervision of each step of the algorithm by a licensed physician is mandatory by law, if the application shall support the decision making process of a diagnosis.

However, the visualizations are restricted by the layout of the web page and the Shiny technology. While Shiny is convenient for building web applications based on R scripts, it is not build for creating powerful visualizations. For the task at hand, i. e. presenting the results of the *ddPCRclust* algorithm and allowing the user to view, modify, and export them, a lot of effort was necessary to customize and extend the base functionalities of the *shiny* library — even though there is only a single algorithm with clearly defined tasks. The presented solution proofs to be sufficient, but for a more comprehensive platform that works with various types of data and allows for different types of analyses with more powerful visualizations, a different technology is necessary. In the next part of this thesis I present *Omics Fusion*, a web based platform build specifically with these goals in mind.

7.4 OUTLOOK

Since the main use of ddPCR at the moment is in the medical field, the GUI could be extended to enrich the results of the *ddPCrclust* algorithms with data from other databases, such as the cancer genome atlas (TCGA) (WEINSTEIN et al., 2013). Differently expressed biomarkers could be highlighted and their risk factor evaluated directly within the *ddPCRvis* software, giving the physician all the information provided within the TCGA.

In addition to that, the genomic location of the biomarkers could be interactively highlighted on a stylized chromosome, enabling researchers for instance to spot regions that behave similarly or areas with especially high mutation frequencies.

Part III

OMICS FUSION — A PLATFORM FOR INTEGRATIVE ANALYSIS OF OMICS DATA

The third part of this thesis presents *Omics Fusion*, a comprehensive software for the integrative analysis and visualization of certain types of biological data. The eighth chapter introduces the motivation and the data, and gives an overview over the software. The ninth chapter presents an example for a task-oriented visualization of functional annotated omics data based on the established Clusters of Orthologous Groups (COG) database and gene ontology (GO) terms. Lastly, the tenth chapter concludes this thesis.

8

INTRODUCING OMICS FUSION

In this chapter, I will introduce *Omics Fusion*, a platform for the integrative analysis of omics data. I will first present the motivation behind this project, before explaining the necessary biological and technical background, e. g. the definition of the term *omics*. Furthermore, I will briefly introduce some implementation details and give an overview over the functionalities. Parts of this chapter have been published under Brink et al. (2016).

8.1 MOTIVATION

With the advance of technology, generating data is no longer the limiting factor in biology. High-throughput experimental technologies transformed biological research from a relatively data-poor discipline to one that is data-rich. A key aspect of understanding and analyzing data is visualization (see Chapter 1). Analytical tools are very useful to solve a specific computational problem, whereas a powerful visualization can enable researchers to gain a mental model for their data and apply their biological knowledge.

Typically, molecular biology strives to understand and potentially optimize metabolic processes within a biological system such as a cell. Cells are living systems full of various functional molecules, which eventually determine the phenotype of the cells. Such molecules include mRNA transcribed from DNA, proteins translated from mRNA, and various metabolites generated by various enzymatic activities. Therefore, only analyzing the DNA sequences of genomes is not sufficient to obtain crucial information regarding the regulatory mechanisms involved in a cell's metabolism, e. g. responses to environmental factors and other stresses, or the production of metabolites. To understand the cell as one system, data from more than one omics discipline is needed (Zhang, Li, and Nie, 2010).

8.2 OMICS TECHNIQUES

To fully understand a biological metabolism and its responses to environmental factors, it is necessary to include functional characterization and accurate quantification of all levels — gene products, proteins, and metabolites; as well as their interaction. In the past decades, significant advancements in improving analytical technologies pertaining to measuring mRNA, proteins, and metabolites have been made. These advancements have led to the generation of new

research fields called omics: genomics, transcriptomics, proteomics, metabolomics, interactomics, and even more advanced fields like fluxomics (determining the rates of metabolic reactions within a biological entity) or localizomics (discovering information about the localization of proteins and metabolites). In general, all experimental approaches that share the following three major features in contrast to traditional procedures can be called omics:

1. Approaches that are high-throughput, data-driven, holistic and top-down methodologies.
2. The attempt to understand the cell metabolism as one integrated system rather than as mere collections of different parts by using information of the relationships between many measured molecular species.
3. The generation of large amounts of data and the analysis of these data often requires significant statistical and computational efforts.

The four major fields in omics, genomics, transcriptomics, proteomics and metabolomics are briefly described below.

8.2.1 *Genomics*

With the completion and publication of the *Haemophilus influenzae* genome sequence in 1995 (Fleischmann et al., 1995) or at the latest after the publication of the Pyrosequencing technology in 2001 (Ronaghi, 2001), DNA sequencing has become the most data-rich field in modern biology and hence genomics was the first of the omics fields. Going back to 1977, when Frederick Sanger published his method for "DNA sequencing with chain-terminating inhibitors" (Sanger, Nicklen, and Coulson, 1977), DNA sequencing has developed from manually sequencing hundreds of basepairs per week to sequencing billions of basepairs in a matter of days (Goodwin, McPherson, and McCombie, 2016). Concurrently, the costs for sequencing have plummeted and the famous "1 000 dollar genome", i. e. the possibility to sequence a human genome for less than a thousand US dollars first anticipated by Mardis (2006), has basically become reality (Figure 8.1). State of the art technologies involve single-molecule real-time sequencing (Pacific Biosciences), ion semiconductor (Ion Torrent sequencing), or sequencing by synthesis (Illumina). However, a sequenced genome doesn't provide any information about the actual gene expression and is more like a set of tools without any information of its actual use. Therefore, a new field called transcriptomics was necessary.

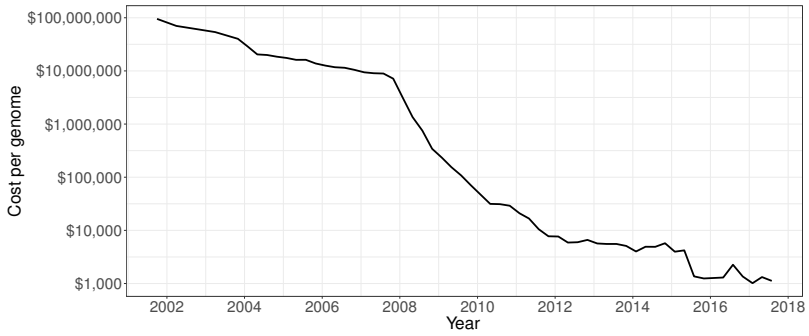


Figure 8.1: The sequencing cost per genome from mid 2001 until mid 2017. The y-axis shows the costs in US dollars on a logarithmic scale. The assumed genome size was 3 000 megabases (i. e. the size of a human genome). The assumed sequence coverage needed depends on the average sequence read length of the sequencing platform; Sanger-based sequencing (average read length = 500–600 bases): 6-fold coverage, Pyrosequencing (average read length = 300–400 bases): 10-fold coverage, Illumina and SOLiD sequencing (average read length = 75–150 bases): 30-fold coverage (WETTERSTRAND, 2017).

8.2.2 *Transcriptomics*

Transcriptomics is the analysis of gene expression and attempts to measure the whole set of all RNA molecules produced in one cell or a population of cells (see Chapter 4). The abundance of these so called transcripts defines the expression level of their corresponding region in the genome, e. g. genes. Studies on individual transcripts have been performed as early as 1979 (SIM et al., 1979). In the 1980s, low-throughput Sanger sequencing began to be used to sequence random individual transcripts from these libraries, called *expressed sequence tags* (PUTNEY, HERLIHY, and SCHIMMEL, 1983; SUTCLIFFE et al., 1982). However, this approach could only evaluate a limited number of genes at a time. In 1995, *serial analysis of gene expression* was the first technology to analyze thousands of transcripts at a time (VELCULESCU et al., 1995). In the same year, the first paper using a DNA microarray was published (SCHENA et al., 1995). With the help of these technologies, in the mid-to-late 1990s countless genome-wide studies have examined the dynamics of gene expression in many model systems and environments, which can be seen as the birth of transcriptomics.

Microarrays became the dominant methodology for almost 10 years, until NAGALAKSHMI et al. (2008) published the transcriptional landscape of the yeast genome by using a new technology called RNA sequencing. It is based on the same technologies that are used for genomics studies, except the RNA is converted to its DNA complement first. Due to the aforementioned increase in throughput and decrease

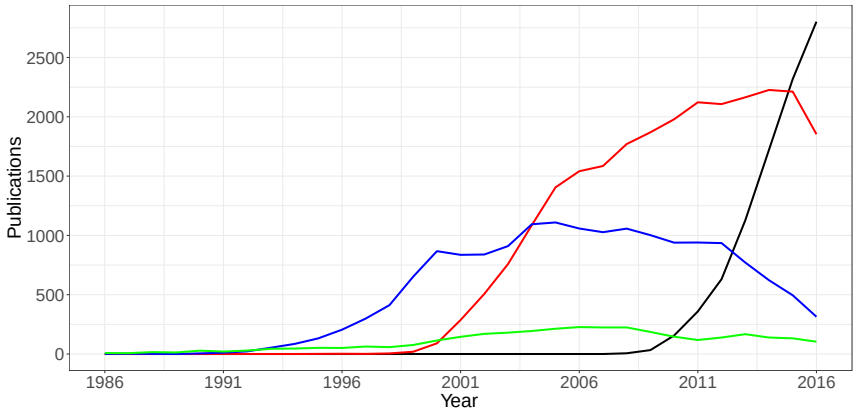


Figure 8.2: The use of transcriptomics methods in the last 30 years. Published papers on PubMed referring to RNA-seq (black), RNA microarray (red), expressed sequence tag (blue) and serial analysis of gene expression (green) since 1986 (*Medline trend*).

in cost when sequencing DNA, RNA sequencing has become the *de facto* standard for transcriptomics studies today (see Figure 8.2).

Furthermore, this data is quantitative, which means the dynamic expression of mRNA molecules and their variation between different states becomes traceable on a genome scale. The resulting data can be easily integrated with genomics data, as both represent the same level - the genes. However, the various possibilities of post-translational modifications are not captured by these analysis and so the integration with proteomics and metabolomics data is more challenging.

8.2.3 *Proteomics*

Proteins are vital parts of living organisms, as they are the major components for building the cellular structure, serve as catalytic enzymes in metabolic pathways, and as signal transduction proteins in regulatory pathways of cells. The term proteomics was coined to make an analogy with genomics and transcriptomics as a tool for the large-scale study of proteins, particularly their functions and structures. However, proteomics is the most difficult of the *omics* fields and even advanced technologies like 2D-PAGE or LC/MS-MS usually cover only 20–40 % of the proteins. Furthermore, all experimental approaches struggle with high error rates and quantitative data on a large scale is not available.

Thus integrating proteomics data with data from other *omics* approaches proves to be very difficult, however additional data from transcriptomics experiments can help to verify proteomics data and reduce the error rates (Zhang, Li, and Nie, 2010).

8.2.4 *Metabolomics*

Metabolites are small molecules that are chemically transformed during metabolism and, as such, they provide a functional readout of cellular state. Unlike genes and proteins, the functions of which are subject to epigenetic regulation and post-translational modifications, respectively, metabolites serve as direct signatures of biochemical activity and are therefore easier to correlate with the phenotype. In this context, metabolite profiling, or metabolomics, is typically performed by employing gas chromatography time-of-flight mass spectrometry, high-performance liquid chromatography-mass spectrometry or capillary electrophoresis-mass spectrometry instruments, nuclear magnetic resonance spectroscopy, and more recently vibrational spectroscopy - or a combination of the above.

The integration with transcriptomics data can be very powerful, since both approaches provide quantitative data and can build the relationship between information elements (genes/transcripts) and functional elements (metabolites). However, special attention is required by the fact that transcriptomics data only provides a relative quantification, whereas metabolomics data includes absolute quantification (PATTI, YANES, and SIUZDAK, 2012).

8.3 RELATED WORK

Many attempts have been made to create software for omics data. Due to the variety and scale of the data, there is no all-in-one solution suitable for every purpose. Here, I present the most notable examples of software in the omics fields and their respective strengths.

CELLDESIGNER CellDesigner is a Java application for Windows, Mac, and Linux developed by FUNAHASHI et al. (2008). It enables users to model gene-regulatory and biochemical networks using the GUI. Networks can be created from scratch or loaded from Systems Biology Markup Language (SBML) files. A layout algorithm creates the initial representation, which can then be edited. This is supported by a number of different annotations for biochemical molecules and their interactions. Networks can be exported to SBML files or converted into Scalable Vector Graphics (SVGs).

CYTOSCAPE Cytoscape is an open source software platform for visualizing molecular interaction networks and biological pathways and integrating these networks with annotations, gene expression profiles and other state data (SHANNON et al., 2003). Although Cytoscape was originally designed for biological research, now it is a general platform for complex network analysis and visualization. The central object in Cytoscape is the network graph, where attributes like

genes, proteins, or other entities are displayed as nodes, while edges represent interactions between those entities. Nodes and edges can be manipulated freely within the GUI. Color, size, and shape of each element can be altered and the network can be searched and filtered according to the current needs. Furthermore, Cytoscape supports the use of plugins (now called Apps), which extend the base features and offer customized solutions for many problems.

OMIX Omix is a commercial tool for creating and editing biochemical networks (DROSTE et al., 2011). The main application field is the interactive mapping of multi-omics data, i. e. the fields of transcriptomics, metabolomics, and fluxomics, onto the network drawing. To achieve this, Omix uses a proprietary scripting language called *Omix Visualization Language*. The visual properties of nodes and edges composing the network can be accessed and modified with these script, allowing users to tailor a visualization to their needs. Data can be imported from and exported to a variety of databases and file formats, including SBML, spreadsheets, bitmap and vector graphic formats, or animations (SWF Flash) and movie files.

3OMICS 3Omics is a web tool for visualizing and integrating multiple inter- or intra-transcriptomic, proteomic, and metabolomic human data (KUO, TIAN, and TSENG, 2013). It covers five commonly used analyses including correlation networks that display the degree of association between variables over multiple time series, clustering of co-expression profiles of different omics data visualized by heat maps, phenotype mapping that uses transcriptomic and proteomic data provided by the user, metabolic pathway enrichment analysis which interprets user provided metabolite data sharing common biological pathways, and GO functional profiling that provides information of cellular components, biological processes and the molecular function of transcriptomic data supplied by the user.

PROMETRA ProMeTra, developed by NEUWEGER et al. (2009), is an open source framework that provides visualization methods for proteomics, metabolomics, and transcriptomics datasets and uses mostly static SVG graphics to generate pathway maps. Additional information about the different omics experiments is added using heatmaps or other color codes. It offers connections to other tools from the Bioinformatics Resource Facility in Bielefeld like MeltDB (metabolomics) (KESSLER, NEUWEGER, and GOESMANN, 2013), Qupe (proteomics) (ALBAUM et al., 2009), and EMMA (transcriptomics) (DONDRUP et al., 2009). Users can upload their own pathways in the form of SVG files, annotate them with additional information for instance from spreadsheets, and download the annotated pathways again as SVGs.

8.4 IMPLEMENTATION

Omics Fusion is a web based platform built on the Spring Web MVC framework for Java. The framework provides model-view-controller (MVC) architecture and ready components that can be used to develop flexible and loosely coupled web applications. The MVC pattern separates three different aspects of the application:

- The model is the central component of the pattern. It directly manages the data, logic and rules of the application.
- The view is responsible for rendering the model data and in general it generates HTML output that the web browser can interpret.
- The Controller is responsible for processing user requests and updating the model and the view appropriately.

The decoupling of these major components allows for efficient code reuse and parallel development.

Omics Fusion was developed as the successor to ProMeTra, an open source framework that provides visualization methods for polyomics datasets and uses static SVG graphics to generate pathway maps. In contrast, *Omics Fusion* focuses on interactivity. By using modern JavaScript techniques, new means for creating powerful visualizations are available, for example allowing users to click on objects, mark certain areas, zoom in or out, etc. This enables them to explore data without prior knowledge about it. Users can apply different filters, evaluate different clustering methods, search for patterns of co-regulated or differentially expressed transcripts, proteins and metabolites, or discover pathways that are affected by a certain condition.

To achieve this high level of interactivity and simplify the implementation, the various visualizations are generated using the powerful *D3* library for Javascript. *D3.js* (or just *D3* for *Data-Driven Documents*) is a JavaScript library created by Bostock, Ogievetsky, and Heer (2011) that uses digital data to drive the creation and control of dynamic and interactive graphical forms which run in web browsers. Embedded within an HTML webpage, the JavaScript *D3.js* library uses pre-built JavaScript functions to select elements, create SVG objects, style them, or add transitions, dynamic effects or tooltips to them. Large datasets can be easily bound to SVG objects using simple *D3* functions to generate rich text/graphic charts and diagrams. This makes it easy for developers to extend the platform and add new visualizations in the future. It is also possible to call external scripts (Python, R, etc.), providing even more ways to customize a workflow.

Data management and manipulation are implemented with Java Servlets. Servlets are Java programming language classes used to extend the capabilities of a server, in contrast to applets, which run locally in web browsers. Therefore, tasks that require lots of computing

power, such as clustering algorithms, can be executed on the server side, and make use of the computational power of the Bioinformatics Resource Facility in Bielefeld.

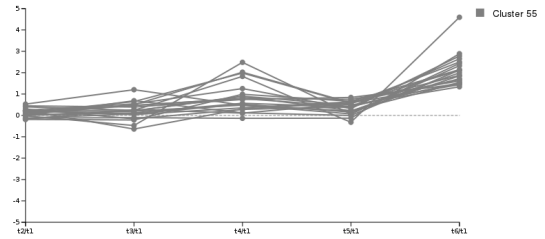
8.5 FUNCTIONALITY

Omics Fusion is a platform for results of all kinds of data-rich high-throughput experiments, focusing on three classical fields of omics studies: transcriptomics, proteomics and metabolomics. It offers convenient data management, such as automated import of spreadsheets, along with connections to other platforms like EMMA (Dondrup et al., 2009), a system for the collaborative analysis and integration of microarray data, MeltDB (Kessler, Neuweger, and Goesmann, 2013), a software platform for the analysis and integration of metabolomics experiment data, or QuPE (Albaum et al., 2009), a rich internet application for the analysis of mass spectrometry-based quantitative proteomics experiments. Here, I present the core functionalities of *Omics Fusion* below in the order of a typical workflow.

8.5.1 *Data manipulation*

There are multiple tools available to manipulate data, ranging from simple but crucial normalization and filtering steps to transformation and missing value replacement. Data can also be enriched by querying other databases like KEGG, UniprotKb or NCBI/Entrez (see Figure 8.3).

Figure 8.3: This figure shows an example screenshot from *Omics Fusion* for a data management screen.



name	descript...	COG	EC	GO	clu	t2...	t3/t1	t4/t1	t5/t1	t6/t1
cg0113 T	-	E	Purine metabolism(3.5.1.5)	-	55	0.1...	0.212	0.885	0.220	1.527
cg2554 T	-	G	Pentose phosphate pathway(2.7.1.15)	-	55	0.0...	0.688	0.121	-0.002	2.118
cg1543 T	-	F	Metabolic pathways(3.2.2.1)	-	55	0.4...	0.393	0.746	0.842	1.475
2-Phospho-D-glycerate H	-	-	Degradation of aromatic compounds	-	55	0.0...	0.201	0.481	0.204	1.993
L-Valine H	-	-	Degradation of aromatic compounds	-	55	-0....	-0.103	-0.136	-0.128	2.888
Phosphoenolpyruvate H	-	-	Degradation of aromatic compounds	-	55	-0....	0.071	0.527	0.158	1.851
cg1784 T	-	E	Biosynthesis of secondary metabolites(4.3.1.12)	-	55	0.1...	0.648	1.983	0.569	2.112
cg0229 T	-	E	Biosynthesis of secondary metabolites(1.4.1.13)	-	55	0.2...	0.541	1.256	0.319	2.779

Figure 8.4: This figure shows an example for cluster profiles. Data points with a similar expression signature are clustered together.

8.5.2 *Data analysis*

Omics Fusion offers tools for descriptive statistics and distribution analysis to get an overview over the data, but also analysis of variance (ANOVA) for robust statistical testing. Besides other classical methods like principial component analysis (PCA), *Omics Fusion* offers a hierarchical cluster analysis with automatic calculation of optimal cluster size and cluster grouping. This hierarchical clustering can be performed on data from multiple *omics* fields, grouping transcriptomic, proteomic, and metabolomic data points with a similar signature. This facilitates the discovery of similar expression patterns throughout experiments from different *omics* fields (see Figure 8.4).

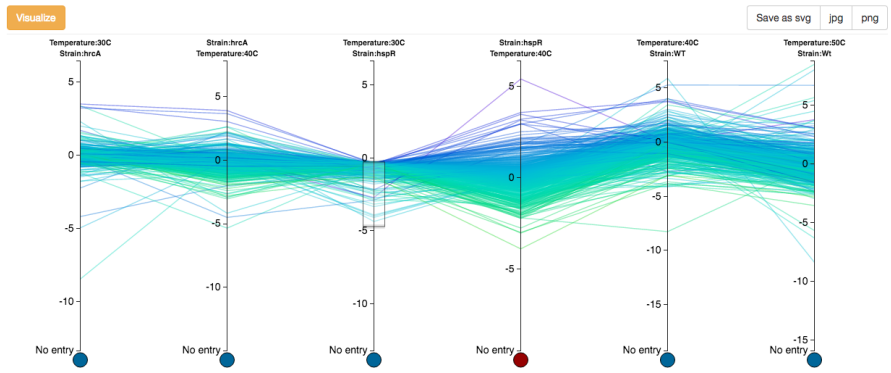


Figure 8.5: This figure shows an example for parallel coordinates. Each axis could correspond to a time point, experimental condition, or strain, each vertical line corresponds to a gene, protein, or metabolite. The users can select an area of interest on each axis to filter the data, as shown here for values between -5 and 0 on the third axis.

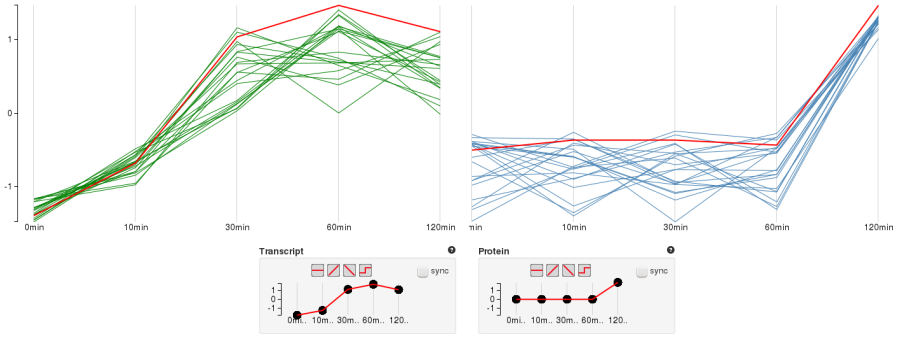


Figure 8.6: This figure shows an example for visual profiling. The bottom shows the desired expression profiles over five time points for transcripts and proteins. The top shows the corresponding data that matches the selected profile.

8.5.3 *Visualization methods*

Omics Fusion offers an increasing number of ways to explore and visualize omics data. A few examples are box plots, scatter plots, parallel coordinates (see Figure 8.5) or parallel sets. Beyond that, users can choose from a number of custom visualizations that introduce new ways to look at data from different omics disciplines. An example for that is a method termed “visual profiling", which allows users to manually draw an arbitrary abundance profile and find all transcripts, proteins, or metabolites matching that prototype (see Figure 8.6).

8.5.4 *Pathway map*

The pathway viewer component implemented within *Omics Fusion* enables the mapping of complete omics datasets on metabolic pathway images. Customized pathway maps can be easily imported as SVG-files and the interactive visualization provides different levels of highlighting important aspects of the data, e. g. stylized icons for different expression patterns or a heatmap representation. The color mapping can be changed to aid color blind people. Individual parts of the visualization such as the names of enzymes or metabolites can be hidden, in order to enhance visibility of the expression patterns. The background can also be changed to black, to further enhance the contrast (see Figure 8.7).

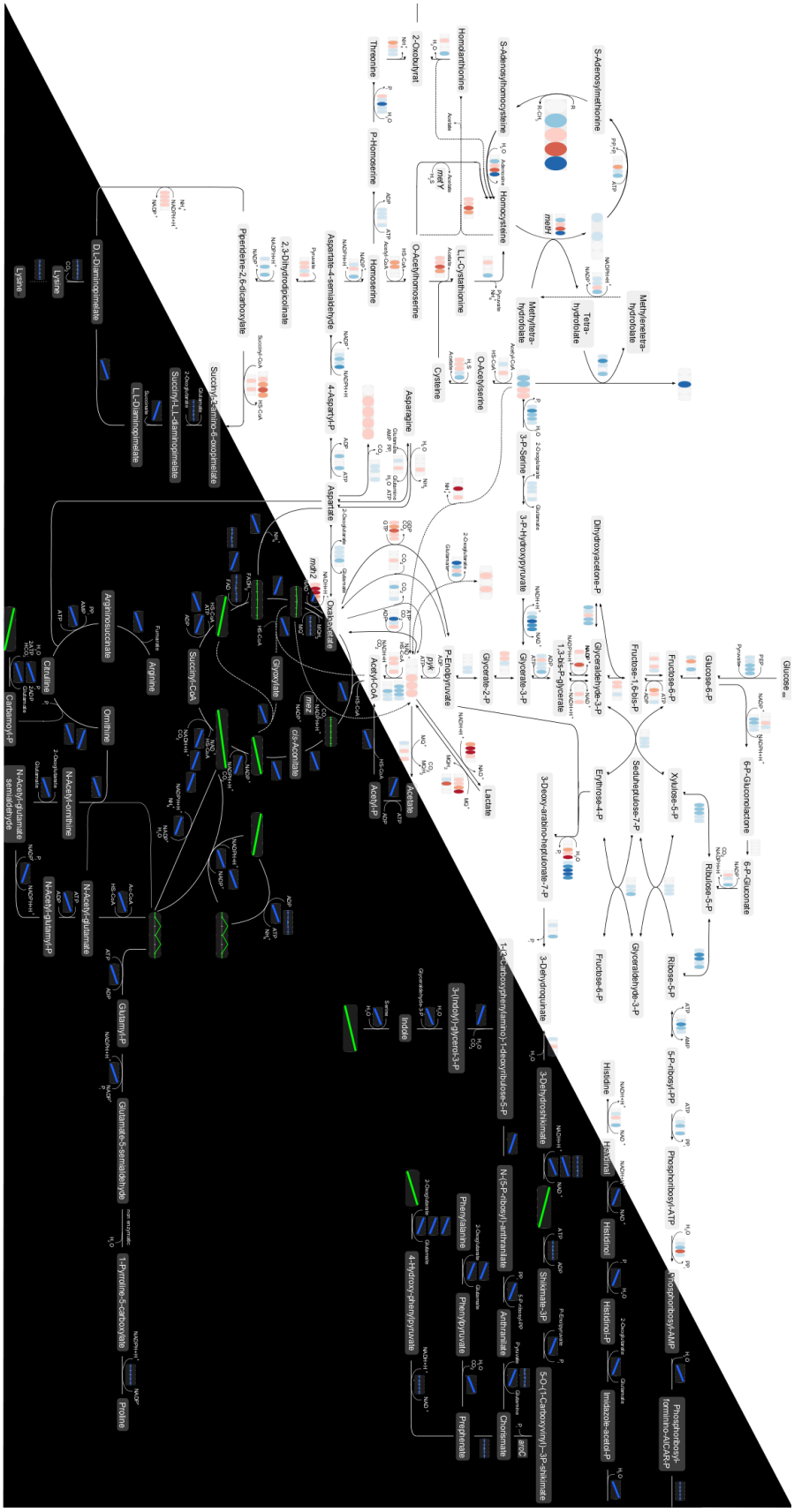


Figure 8.7: This figure shows an example for a pathway map, presenting two different visual representations of the same pathway.

9 VISUALIZATION OF FUNCTIONAL ANNOTATION DATA

In this chapter, I will present a task-oriented visualization approach of functional annotated omics data based on the established Clusters of Orthologous Groups (COG) database and gene ontology (GO) terms.

9.1 FUNCTIONAL ANNOTATION

In the context of this thesis, functional annotation refers to attaching information regarding a biological function to a gene, gene product, or protein. Subsequent, I present two approaches for this.

9.1.1 *Gene Ontology*

In computer science and information science, an ontology is a formal naming and definition of the types, properties, and interrelationships of the entities that really exist in a particular domain of discourse. Thus, it is a practical application of philosophical ontology, with a taxonomy. In bioinformatics, the gene ontology initiative aims to establish a unified ontology for genes and gene product attributes across all species (Ashburner et al., 2000). Each GO term has a unique id and can have one or multiple parent terms. In the following example, the term `GO:0000100` has the parents `GO:0072349` and `GO:1901682`.

```
[Term]
id: GO:0000100
name: S-methylmethionine transmembrane transporter activity
namespace: molecular_function
alt_id: GO:0015178
def: "Enables the transfer of S-methylmethionine from one side
        of a membrane to the other." [GOC:ai]
subset: gosubset_prok
synonym: "S-methylmethionine permease activity" EXACT []
synonym: "S-methylmethionine transporter activity" BROAD []
is_a: GO:0072349 ! modified amino acid transmembrane transporter activity
is_a: GO:1901682 ! sulfur compound transmembrane transporter activity
```

9.1.2 *Clusters of Orthologous Groups*

In biology, the term homology is defined as biological structures that descended from a common ancestor. With the advance of genome sequencing, it became possible to compare genomes and extend this definition to DNA sequences. Orthologous sequences are defined as

homologous sequences that were separated by a speciation event. Orthologs typically have the same function, allowing scientists to infer the function of potentially unknown genes or proteins. TATUSOV, KOONIN, and LIPMAN (1997) proposed a system to categorize DNA sequence similarities into Clusters of Orthologous Groups (COG). Today, the COG database contains 4 631 COGs for prokaryotes subdivided into 26 functional categories (GALPERIN et al., 2014). The growing number of eukaryotic genomes being sequenced has derived a similar database for eukaryotes under the name of eukaryotic orthologous groupss (KOGs) (TATUSOV et al., 2003).

9.2 SEMANTIC REASONER

A semantic reasoner can be used to model Description Logic ontologies, such as GO. The difficulties in constructing such models primarily arise from two sources. First, there are often a great number of different possible constructions and second, the models built by tableau reasoners can be extremely large, even for relatively small ontologies. The HermiT reasoner by GLIMM et al. (2014) provides an efficient implementation of semantic reasoning based on a novel hypertableau calculus (MOTIK, SHEARER, and HORROCKS, 2009). In *Omics Fusion*, I use HermiT to construct the GO ontology, query it for GO terms, find parent and child nodes, etc.

9.3 THE VISUALIZATION

According to Chapter 1, the following questions have been answered in the design process of the visualization. First, the data type needed to be defined. Here, the available data type is tables, containing expression values for genes or proteins, which are further categorized in different sets based on their functional annotation. Second, the task needs to be defined, which in this case is discovery, e. g. discovering which set has the highest/lowest variance or looking for outliers. The target can be the whole data or a subset. Third, a visual idiom needs to be designed. As an appropriate encoding I choose to map the data to box-and-whisker plots.

9.3.1 *Box-and-whisker plots*

Box-and-whisker plots, or simply box plots, are a visual encoding for numerical values using shape and size. It is based on the quartiles of the data, i. e. the three points that divide the data set into four equal groups, each group comprising a quarter of the data. The quartiles are encoded into boxes, where the lower and the upper quartile define the borders. The median is usually marked with a line inside the box. The whiskers can be encoded in different ways. In this visualiza-

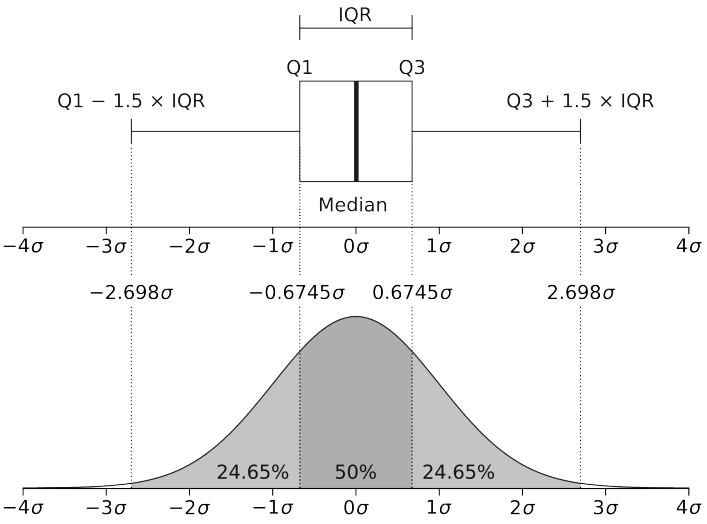


Figure 9.1: Boxplot with an interquartile range and a probability density function of a Normal $N(0, \sigma^2)$ Population.
Adapted from: `https://commons.wikimedia.org/wiki/File:Boxplot_vs_PDF.svg`

tion, I follow the description by McGill, Tukey, and Larsen (1978), sometimes also called Tukey boxplot. They define the end points of the whiskers as the lowest and highest data point still within 1.5 interquartile range (IQR) from the lower and upper quartile, respectively (see Figure 9.1). Data points outside this range can be added as individual points to show outliers.

Box plots are useful to quickly grasp the behavior of groups or sets of values graphically. They take little space and highlight both the median and the variance of the data, which makes it easy to compare groups or sets of values or discover outliers.

9.3.2 *COG/GO box plots*

In *Omics Fusion*, box plots can be generated for functional annotated transcriptomic or proteomic data. The data comprises the relative expression of genes or proteins versus a control (e. g. the first time point or a standard condition).

Each visualization in *Omics Fusion* can be configured with a control panel on the top of the page. The control panel for this visualization is shown in Figure 9.2. The users can select, how many of the datasets that are saved in the database for this experiment they want to include in the visualization. Each dataset corresponds to a time point, experimental condition, or strain. They also have to select if they want to visualize transcript or proteins, and whether they want to create the visualization based COG or GO terms.

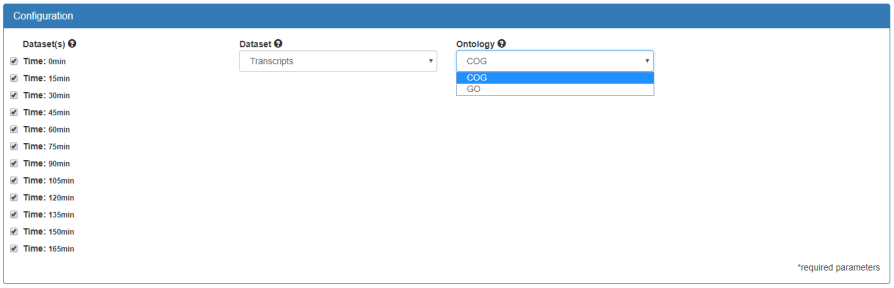


Figure 9.2: Screenshot of a configuration panel for a COG/GO box plot. The users can select, how many datasets they want to include in the visualization, which dataset they want to visualize (i. e. transcripts or proteins), and whether they want to create the visualization based COG or GO terms.

The resulting plot is shown in Figure 9.3. Each box plot represents a functional COG category. The color of the box reminds the users, which data type they selected. Green corresponds to transcriptomics, blue to proteomics, and orange to metabolomics data. This encoding is consistent throughout *Omics Fusion*. The users can move the mouse over a box to get more details about the category, e. g. category "N" refers to "cell motility" as seen in Figure 9.3. If the users select GO terms instead of COGs, they can also click on a box and the visualization will change to show the children of the selected GO term. This way, users can filter the data and narrow down the most (or least) differently expressed genes. Furthermore, each visualization can be exported as a svg, png, or jpg file.

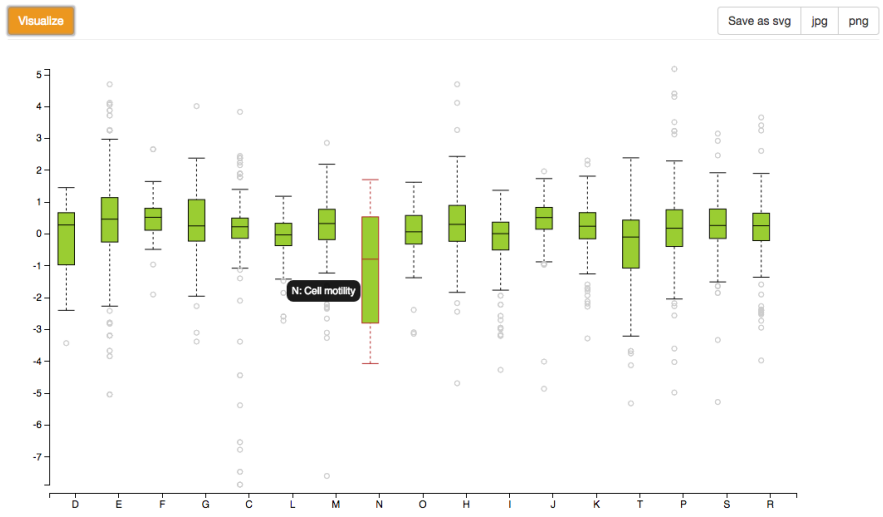


Figure 9.3: Screenshot of a COG box plot based on transcriptomics data obtained from *Caulobacter crescentus*. The y-axis represents the gene expression, the x-axis shows the COG categories. On mouse over, details for these categories are shown, as demonstrated here with category "N".

9.4 DISCUSSION

Omics Fusion is an extendible, web-based platform for the integrative analysis of omics data. It provides powerful analysis tools, including established methods for analyzing and visualizing single omics data, as well as new features for an integrative analysis of data from multiple omics disciplines, which can potentially provide new insights into biology, or at least simplify gathering of information and analyzing data from experiments with more than one omics approach.

Compared to other tools that offer a similar level of interactivity, *Fusion* does not solely focus on networks and pathways (e. g. CellDesigner, Omix, or Cytoscape), nor is it limited to a specific organism (e. g. 3Omics). Instead, *Omics Fusion* puts an emphasis on visualization and data exploration.

The presented example of a visualization of functional annotated data based on the established Clusters of Orthologous Groups (COG) database and gene ontology (GO) terms shows how an interactive visualization can facilitate knowledge discovery. Adding the functional annotation layer to the visualization creates a connection between the data and its biological meaning. The users can visualize their expression data based on COGs, in order to get an overview over what is happening in a cell or organism at the selected condition or time point. Categories that are differentially expressed can be spotted easily, such as the "cell motility" category in Figure 9.3. This transcriptomics data was obtained from *Caulobacter crescentus*. This bacteria is a model organism for studying the regulation of the cell cycle. It can have two different forms — a swarmer cell, which that has a single flagellum at one cell pole that provides swimming motility for chemotaxis; or a stalked cell that has a tubular stalk structure with an adhesive holdfast material on its end, with which the stalked cell can adhere to surfaces. Hence, genes belonging to the "cell motility" category are of high interest in this experiment.

The additional functionality of navigating through the GO terms, this way only looking at subsets of the data and finding the most (or least) differently expressed genes can potentially lead to unexpected discoveries that might have gone unnoticed otherwise.

In contrast to the previous examples, the visualizations in *Omics Fusion* need to be designed so they work with a variety of data and tasks. This presents a challenge to the design process. In part II of the thesis, the problem was clearly defined as identifying CNAs in order to predict cancer progression. Here, the tasks are abstract and visualizations need to be designed on that basis. In the case of the COG/GO box plot, the visualization is only useful if the users are looking for the overall variance in gene or protein expression. If they want to compare multiple time points or conditions, the visualization is rendered useless. Hence, *Omics Fusion* offers a broad selection of

tools and visualizations in order to be able fulfill the users needs. This requires much more time and effort, with over 13 people contributing to the software over time, compared to only one person in the case of *ddPCRvis* (see Chapter 7).

9.5 OUTLOOK

Omics Fusion has the potential to develop into a widely accepted platform for integrative analysis of omics data. New tools and visualizations are added continuously. An extensive study with a comprehensive dataset as a use case could show the benefits of such a platform to the community and potentially have a big impact.

For the existing tools such as the COG/GO box plot, a usability study would give insights into how well they are designed and where the design could be enhanced. Further improvements such as reducing the required time for the platform to create the visualization could be desirable.

10 CONCLUSION

In this thesis, I explored different aspects of analysis and visualization of scientific data, particularly data from biology and biochemistry. After introducing the general concepts of data mining and visualization, I presented a static, non interactive visualization of amino acids in the form of a physical card game, in order to facilitate memorizing the amino acids and some of their important physicochemical properties. The card game provides a viable solution as a learning aid and has been received favorably. Card games have a long tradition and their public familiarity lowers the learning curve for this visualization. The fun factor aids both the process of memorizing as well as overcoming the initial reluctance to start learning the new thing. Furthermore, the physical print version of the cards makes it possible to take them anywhere and use them anytime, in contrast to a computer based visualization.

In Part II of the thesis, I tackled a problem from the field of cancer diagnostics. First, I explained the biological background of the problem. Then, I presented an R package (ddPCRclust) and accompanying shiny interface (ddPCRvis). I showed that the results of *ddPCRclust* are on par with manual annotation by experts, while the computation only takes a few minutes per 96-well experiment. Three independent clustering approaches provide robustness, which is especially important in a medical context. The web based GUI *ddPCRvis* does not only provides easy access to the algorithm, but it also enables the user to check the results and manually correct them if necessary. Furthermore, in a medical context supervision of each step of the algorithm by a licensed physician is mandatory by law, if the application shall support the decision making process of a diagnosis. However, the visualizations are restricted by the layout of the web page and the shiny technology.

Part III of the thesis is situated in the field of biology. I presented *Omics Fusion*, a comprehensive software for the integrative analysis and visualization of omics data. I described the background and the motivation behind the project and compared it to other state of the art tools. I detailed the implementation and functionalities of *Omics Fusion*, before presenting an example of a visualization based on functional annotated expression data. The visualization is based on COGs or GO terms and can be used in order to get an overview over what is happening in a cell or organism at the selected condition or time point. Categories that are differentially expressed can be spotted easily. In contrast to the previous examples, the visualizations in *Omics*

Fusion needed to be designed so they work with a variety of data and tasks. This required much more time and effort, with over 13 people contributing to the software over time, compared to only one person in the case of *ddPCRvis*.

To summarize, I could show that data mining and visualization are disciplines of great significance. They have become a part of many aspects today's life, including scientific research. I presented examples from teaching, cancer research, and molecular biology, each requiring its own, task oriented approach. The solutions appear very different (e. g. a card game and a web software), but the design process was similar and can serve as a guideline for future visualizations.

We have seen that computer programming is an art,
because it applies accumulated knowledge to the world,
because it requires skill and ingenuity, and especially
because it produces objects of beauty.

— Donald E. Knuth, 1974

INDIVIDUAL CONTRIBUTIONS

The following authors contributed to Part I of this thesis:

Benedikt G. Brink wrote and revised the content. Georges Hattab conceived the amino acids task-oriented visualization, led its development, designed the amino acids cards and the 2-dimensional structural formulae. Benedikt G. Brink suggested an appropriate data abstraction, implemented the gamification approach, and contributed to the execution of the data abstraction design. Tamara Munzner refined the description of the visual encoding of the cards. Tim W. Nattkemper aided the explication of the motivation behind the visualization.

The following authors contributed to Part II of this thesis:

Benedikt G. Brink wrote and revised the content. Tim W. Nattkemper guided the revision. Benedikt G. Brink developed *ddPCRclust* and *ddPCRvis*. Justin Meskas implemented the initial prototype of the flowDensity based clustering approach. Benedikt G. Brink extended this approach, suggested and implemented the other two approaches, as well as the shiny interface. X. J. David Lu and Curtis B. Hughesman provided the data and helped testing the software. Ryan R. Brinkman conceived the project and supervised the development process.

The following authors contributed to Part III of this thesis:

Benedikt G. Brink wrote and revised the content, implemented the semantic reasoner, the Box-and-whisker plot visualization, and the bar chart visualization. Annika Seidel, Nils Kleinbölting, Sonja Klingberg, Daniel Blume, Peter Belmann, Denis Kramer, Julia Gierens, Stefan Biermann, Yannic Kerkhoff, Carina Wenzel, and Ruben Christian Hamann implemented various modules for *Omics Fusion*. Tim W. Nattkemper supervised many of the visualization modules. Stefan P. Albaum conceived *Omics Fusion* and led its development.

BIBLIOGRAPHY

Acinas, Silvia G, Ramahi Sarma-Rupavtarm, Vanja Klepac-Ceraj, and Martin F Polz (2005). "PCR-induced sequence artifacts and bias: insights from comparison of two 16S rRNA clone libraries constructed from the same sample." In: *Applied and environmental microbiology* 71.12, pp. 8966–8969 (cit. on p. 36).

Aghaeepour, Nima, Greg Finak, Holger Hoos, Tim R Mosmann, Ryan Brinkman, Raphael Gottardo, Richard H Scheuermann, FlowCAP Consortium, DREAM Consortium, et al. (2013). "Critical assessment of automated flow cytometry data analysis techniques." In: *Nature methods* 10.3, pp. 228–238 (cit. on p. 47).

Albaum, Stefan P, Heiko Neuweger, Benjamin Fränzel, Sita Lange, Dominik Mertens, Christian Trötschel, Dirk Wolters, Jörn Kalinowski, Tim W Nattkemper, and Alexander Goesmann (2009). "Qupe—a Rich Internet Application to take a step forward in the analysis of mass spectrometry-based quantitative proteomics experiments." In: *Bioinformatics* 25.23, pp. 3128–3134 (cit. on pp. 78, 80).

Arnheim, Rudolf (1949). "The Gestalt theory of expression." In: *Psychological review* 56.3, p. 156 (cit. on p. 22).

— (1956). *Art and visual perception: A psychology of the creative eye*. Univ of California Press (cit. on p. 22).

Ashburner, Michael, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. (2000). "Gene Ontology: tool for the unification of biology." In: *Nature genetics* 25.1, pp. 25–29 (cit. on p. 85).

Attali, Dean, Roza Bidshahri, Charles Haynes, and Jennifer Bryan (2016). "ddpcr: an R package and web application for analysis of droplet digital PCR data." In: *F1000Research* 5 (cit. on p. 43).

Beaver, Julia A, Danijela Jelovac, Sasidharan Balukrishna, Rory L Cochran, Sarah Croessmann, Daniel J Zabransky, Hong Yuen Wong, Patricia Valda Toro, Justin Cidado, Brian G Blair, et al. (2014). "Detection of cancer DNA in plasma of patients with early-stage breast cancer." In: *Clinical cancer research* 20.10, pp. 2643–2650 (cit. on p. 59).

Berman, Helen M, John Westbrook, Zukang Feng, Gary Gilliland, T N Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne (2000). "The Protein Data Bank." In: *Nucleic Acids Research* 28, pp. 235–242 (cit. on p. 19).

Beroukhim, Rameen, Craig H Mermel, Dale Porter, Guo Wei, Soumya Raychaudhuri, Jerry Donovan, Jordi Barretina, Jesse S Boehm, Jennifer Dobson, Mitsuyoshi Urashima, et al. (2010). "The landscape of somatic copy-number alteration across human cancers." In: *Nature* 463.7283, p. 899 (cit. on p. 33).

Bertram, John S (2000). "The molecular biology of cancer." In: *Molecular aspects of medicine* 21.6, pp. 167–223 (cit. on p. 29).

Bitterman, Morton E (1965). "Phyletic differences in learning." In: *American Psychologist* 20.6, p. 396 (cit. on p. 18).

Bodmer, Walter Fred et al. (1986). *The public understanding of science*. Birkbeck College London, England (cit. on p. 17).

Bostock, Michael, Vadim Ogievetsky, and Jeffrey Heer (2011). "D^3 data-driven documents." In: *IEEE transactions on visualization and computer graphics* 17.12, pp. 2301–2309 (cit. on p. 79).

Brecher, Jonathan (2006). "Graphical representation of stereochemical configuration (IUPAC Recommendations 2006)." In: *Pure and applied chemistry* 78.10, pp. 1897–1970 (cit. on p. 17).

Brink, Benedikt G, Justin Meskas, and Ryan R Brinkman (2018). "ddPCRclust: an R package and Shiny app for automated analysis of multiplexed ddPCR data." In: *Bioinformatics* (cit. on pp. 15, 43).

Brink, Benedikt G, Annica Seidel, Nils Kleinbölting, Tim W Nattkemper, and Stefan P Albaum (2016). "Omics Fusion–A Platform for Integrative Analysis of Omics Data." In: *Journal of integrative bioinformatics* 13.4, pp. 43–46. doi: 10.2390/biecoll-jib-2016-296 (cit. on pp. 15, 73).

Card, Stuart K, Jock D Mackinlay, and Ben Shneiderman (1999). *Readings in information visualization: using vision to think*. Morgan Kaufmann (cit. on p. 13).

Categorical Colours. url: http://bl.ocks.org/aaizemberg/raw/78bd3dade9593896a59d/ (visited on 10/07/2016) (cit. on p. 21).

Chang, Dempsey, Keith V Nesbitt, and Kevin Wilkins (2007). "The Gestalt principles of similarity and proximity apply to both the haptic and visual grouping of elements." In: *Proceedings of the eight Australasian conference on User interface-Volume 64*. Australian Computer Society, Inc., pp. 79–86 (cit. on p. 22).

Chang, Winston, Joe Cheng, JJ Allaire, Yihui Xie, and Jonathan McPherson (2017). *shiny: Web Application Framework for R*. R package version 1.0.3. url: https://CRAN.R-project.org/package=shiny (cit. on pp. 43, 61).

Cher, Michael L, Donal Macgrogan, Robert Bookstein, James A Brown, Robert B Jenkins, and Ronald H Jensen (1994). "Comparative genomic hybridization, allelic imbalance, and fluorescence in situ hybridization on chromosome 8 in prostate cancer." In: *Genes, Chromosomes and Cancer* 11.3, pp. 153–162 (cit. on p. 34).

Chiu, Anthony, Mahmood Ayub, Caroline Dive, Ged Brady, and Crispin J Miller (2017). "twoddpcr: An R/Bioconductor package and Shiny app for Droplet Digital PCR analysis." In: *Bioinformatics*, btx308 (cit. on p. 43).

Cleton-Jansen, Anne-Marie, Elna W Moerland, Net J Kuipers-Dijkshoorn, Cees J Cornelisse, Peter Devilee, David F Callen, Grant R Sutherland, and Bettine Hansen (1994). "At least two different regions are involved in allelic imbalance on chromosome arm 16q in breast cancer." In: *Genes, Chromosomes and Cancer* 9.2, pp. 101–107 (cit. on p. 34).

Darwin, Charles (1859). *On the origin of species by means of natural selection: or the preservation of favoured races in the struggle for life*. John Murray, Albemarle Street (cit. on p. 10).

Dayhoff, Margaret O (1965). "Atlas of protein sequence and structure." In: (cit. on p. 21).

Dobnik, David, Dejan Štebih, Andrej Blejec, Dany Morisset, and Jana Žel (2016). "Multiplex quantification of four DNA targets in one reaction with Bio-Rad droplet digital PCR system for GMO detection." In: *Scientific reports* 6, p. 35451 (cit. on p. 40).

Dondrup, Michael, Stefan P Albaum, Thasso Griebel, Kolja Henckel, Sebastian Jünemann, Tim Kahlke, Christiane K Kleindt, Helge Küster, Burkhard Linke, Dominik Mertens, et al. (2009). "EMMA 2–a MAGE-compliant system for the collaborative analysis and integration of microarray data." In: *BMC bioinformatics* 10.1, p. 50 (cit. on pp. 78, 80).

Droste, Peter, Stephan Miebach, Sebastian Niedenführ, Wolfgang Wiechert, and Katharina Nöh (2011). "Visualizing multi-omics data in metabolic networks with the software Omix—a case study." In: *Biosystems* 105.2, pp. 154–161 (cit. on p. 78).

Dube, Simant, Jian Qin, and Ramesh Ramakrishnan (2008). "Mathematical analysis of copy number variation in a DNA sample using digital PCR on a nanofluidic device." In: *PloS one* 3.8, e2876 (cit. on p. 37).

Dulek, Daniel (2015). *Molecules – a chemistry card game*. URL: `http://playefg.com/` (cit. on p. 18).

Ferlay, Jacques, Isabelle Soerjomataram, Rajesh Dikshit, Sultan Eser, Colin Mathers, Marise Rebelo, Donald Maxwell Parkin, David Forman, and Freddie Bray (2015). "Cancer incidence and mortality worldwide: sources, methods and major patterns in GLOBOCAN 2012." In: *International journal of cancer* 136.5 (cit. on p. 34).

Fleischmann, Robert D, Mark D Adams, Owen White, Rebecca A Clayton, Ewen F Kirkness, Anthony R Kerlavage, Carol J Bult, Jean-Francois Tomb, Brian A Dougherty, Joseph M Merrick, et al. (1995). "Whole-genome random sequencing and

assembly of Haemophilus influenzae Rd." In: *science*, pp. 496–512 (cit. on p. 74).

Funahashi, Akira, Yukiko Matsuoka, Akiya Jouraku, Mineo Morohashi, Norihiro Kikuchi, and Hiroaki Kitano (2008). "CellDesigner 3.5: a versatile modeling tool for biochemical networks." In: *Proceedings of the IEEE* 96.8, pp. 1254–1265 (cit. on p. 77).

Galperin, Michael Y, Kira S Makarova, Yuri I Wolf, and Eugene V Koonin (2014). "Expanded microbial genome coverage and improved protein family annotation in the COG database." In: *Nucleic acids research* 43.D1, pp. D261–D269 (cit. on p. 86).

Ge, Yongchao and Stuart C Sealfon (2012). "flowPeaks: a fast unsupervised clustering for flow cytometry data via K-means and density peak finding." In: *Bioinformatics* 28.15, pp. 2052–2058 (cit. on p. 48).

Glimm, Birte, Ian Horrocks, Boris Motik, Giorgos Stoilos, and Zhe Wang (2014). "HermiT: an OWL 2 reasoner." In: *Journal of Automated Reasoning* 53.3, pp. 245–269 (cit. on p. 86).

Goodwin, Sara, John D McPherson, and W Richard McCombie (2016). "Coming of age: ten years of next-generation sequencing technologies." In: *Nature Reviews Genetics* 17.6, pp. 333–351 (cit. on p. 74).

Halling, Kevin C, Amy J French, Shannon K McDonnell, Lawrence J Burgart, Daniel J Schaid, Brett J Peterson, Laurie Moon-Tasson, Michelle R Mahoney, Daniel J Sargent, Michael J O'connell, et al. (1999). "Microsatellite instability and 8p allelic imbalance in stage B2 and C colorectal cancers." In: *Journal of the National Cancer Institute* 91.15, pp. 1295–1303 (cit. on p. 34).

Hanahan, Douglas and Robert A Weinberg (2011). "Hallmarks of cancer: the next generation." In: *cell* 144.5, pp. 646–674 (cit. on pp. 33, 34).

Hattab, Georges, Benedikt G Brink, and Tim W Nattkemper (June 2016). "A mnemonic card game for your amino acids." In: *Information+ Conference*. DOI: 10.5281/zenodo.55101 (cit. on pp. 14, 17).

Heid, Christian A, Junko Stevens, Kenneth J Livak, and P Mickey Williams (1996). "Real time quantitative PCR." In: *Genome research* 6.10, pp. 986–994 (cit. on p. 36).

Hering, Ewald (1878). *Zur lehre vom lichtsinne*. Vol. 68. K. Akademie der Wissenschaften (cit. on p. 10).

Heyrovska, Raji (2008). "Atomic structures of all the twenty essential amino acids and a tripeptide, with bond lengths as sums of atomic covalent radii." In: *arXiv preprint arXiv:0804.2488* (cit. on p. 21).

Hindson, Benjamin J, Kevin D Ness, Donald A Masquelier, Phillip Belgrader, Nicholas J Heredia, Anthony J Makarewicz, Isaac J Bright, Michael Y Lucero, Amy L Hiddessen, Tina C Legler, et al. (2011). "High-throughput droplet digital PCR system for absolute quantitation of DNA copy number." In: *Analytical chemistry* 83.22, pp. 8604–8610 (cit. on pp. 37–39).

Holmstrup, P, P Vedtofte, Jesper Reibel, and K Stoltze (2007). "Oral premalignant lesions: is a biopsy reliable?" In: *Journal of oral pathology & medicine* 36.5, pp. 262–266 (cit. on p. 35).

Hornik, Kurt (2005). "A clue for cluster ensembles." In: *Journal of Statistical Software* 14.12, pp. 1–25 (cit. on p. 51).

Hubert, Lawrence and Phipps Arabie (1985). "Comparing partitions." In: *Journal of classification* 2.1, pp. 193–218 (cit. on pp. 52, 54).

Hughesman, Curtis B, XJ David Lu, Kelly YP Liu, Yuqi Zhu, Catherine F Poh, and Charles Haynes (2016). "A Robust Protocol for Using Multiplexed Droplet Digital PCR to Quantify Somatic Copy Number Alterations in Clinical Tissue Specimens." In: *PloS one* 11.8, e0161274 (cit. on pp. vii, ix, 40, 46, 59).

Irwin, Alan and Brian Wynne (2003). *Misunderstanding science?: the public reconstruction of science and technology*. Cambridge University Press (cit. on p. 17).

Kessler, Nikolas, Heiko Neuweger, Alexander Goesmann, et al. (2013). "MeltDB 2.0–advances of the metabolomics software system." In: *Bioinformatics* 29.19, pp. 2452–2459 (cit. on pp. 78, 80).

Knuth, Donald E. (1974). "Computer Programming as an Art." In: *Communications of the ACM* 17.12, pp. 667–673 (cit. on p. 93).

Kuo, Tien-Chueh, Tze-Feng Tian, and Yufeng Jane Tseng (2013). "3Omics: a web-based systems biology tool for analysis, integration and visualization of human transcriptomic, proteomic and metabolomic data." In: *BMC systems biology* 7.1, p. 64 (cit. on p. 78).

Larson, Pamela S, Antonio de las Morenas, Sheila R Bennett, L Adrienne Cupples, and Carol L Rosenberg (2002). "Loss of heterozygosity or allele imbalance in histologically normal breast epithelium is distinct from loss of heterozygosity or allele imbalance in co-existing carcinomas." In: *The American journal of pathology* 161.1, pp. 283–290 (cit. on p. 34).

Mahalanobis, Prasanta Chandra (1936). "On the generalised distance in statistics." In: *Proceedings of the National Institute of Sciences of India*, pp. 49–55 (cit. on p. 51).

Malek, Mehrnoush, Mohammad Jafar Taghiyar, Lauren Chong, Greg Finak, Raphael Gottardo, and Ryan R Brinkman (2015). "flowDensity: reproducing manual gating of flow cytometry data

by automated density-based cell population identification." In: *Bioinformatics* 31.4, pp. 606–607 (cit. on p. 47).

Mardis, Elaine R (2006). "Anticipating the $1,000 genome." In: *Genome biology* 7.7, p. 112 (cit. on p. 74).

McDermott, Geoffrey P, Duc Do, Claudia M Litterst, Dianna Maar, Christopher M Hindson, Erin R Steenblock, Tina C Legler, Yann Jouvenot, Samuel H Marrs, Adam Bemis, et al. (2013). "Multiplexed target detection using DNA-binding dye chemistry in droplet digital PCR." In: *Analytical chemistry* 85.23, pp. 11619–11627 (cit. on p. 40).

McGill, Robert, John W Tukey, and Wayne A Larsen (1978). "Variations of box plots." In: *The American Statistician* 32.1, pp. 12–16 (cit. on pp. 68, 87).

Medline trend. url: `http://dan.corlan.net/medline-trend.html` (visited on 11/25/2017) (cit. on p. 76).

Motik, Boris, Rob Shearer, and Ian Horrocks (2009). "Hypertableau reasoning for description logics." In: *Journal of Artificial Intelligence Research* 36.1, pp. 165–228 (cit. on p. 86).

Munzner, Tamara (2014). *Visualization analysis and design*. CRC press (cit. on pp. 7, 9, 11).

Myers, Jeffrey (2009). *Oral cancer metastasis*. Springer Science & Business Media (cit. on p. 35).

Nagalakshmi, Ugrappa, 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." In: *Science* 320.5881, pp. 1344–1349 (cit. on p. 75).

Nelson, David L, Albert L Lehninger, and Michael M Cox (2008). *Lehninger principles of biochemistry*. Macmillan (cit. on p. 19).

Neuweger, Heiko, Marcus Persicke, Stefan P Albaum, Thomas Bekel, Michael Dondrup, Andrea T Hüser, Jörn Winnebald, Jessica Schneider, Jörn Kalinowski, and Alexander Goesmann (2009). "Visualizing post genomics data-sets on customized pathway maps by ProMeTra–aeration-dependent gene expression and metabolism of Corynebacterium glutamicum as an example." In: *BMC systems biology* 3.1, p. 82 (cit. on p. 78).

Ng, David, Derek Tan, et al. (2010). *The PHYLO(MON) project*. url: `http://phylogame.org/` (cit. on p. 17).

Ohno, Susumu (1972). "So much" junk" DNA in our genome." In: 23, pp. 366–370 (cit. on p. 32).

Oxnard, Geoffrey R, Cloud P Paweletz, Yanan Kuang, Stacy L Mach, Allison O'Connell, Melissa M Messineo, Jason J Luke, Mohit Butaney, Paul Kirschmeier, David M Jackman, et al. (2014). "Noninvasive detection of response and resistance in EGFR-mutant lung cancer using

quantitative next-generation genotyping of cell-free plasma DNA." In: *Clinical cancer research* 20.6, pp. 1698–1705 (cit. on p. 59).

Papadimitriou, Christos H and Kenneth Steiglitz (1982). *Combinatorial optimization: algorithms and complexity*. Courier Corporation (cit. on p. 49).

Partridge, M, G Emilion, S Pateromichelakis, R A'Hern, E Phillips, and J Langdon (1998). "Allelic imbalance at chromosomal loci implicated in the pathogenesis of oral precancer, cumulative loss and its relationship with progression to cancer." In: *Oral oncology* 34.2, pp. 77–83 (cit. on p. 34).

Patti, Gary J, Oscar Yanes, and Gary Siuzdak (2012). "Metabolomics: the apogee of the omic triology." In: *Nature reviews. Molecular cell biology* 13.4, p. 263 (cit. on p. 77).

Playfair, William (1801). *The commercial and political atlas: representing, by means of stained copper-plate charts, the progress of the commerce, revenues, expenditure and debts of england during the whole of the eighteenth century*. T. Burton (cit. on pp. 4, 13).

Putney, Scott D, Walter C Herlihy, and Paul Schimmel (1983). "A new troponin T and cDNA clones for 13 different muscle proteins, found by shotgun sequencing." In: *Nature* 302.5910, pp. 718–721 (cit. on p. 75).

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria. URL: `https://www.R-project.org/` (cit. on p. 43).

Raghupathi, Wullianallur (2016). "Data mining in healthcare." In: *Healthcare Informatics: Improving Efficiency through Technology, Analytics, and Management*, pp. 353–372 (cit. on p. 14).

Römpp's Chemistry Lexicon. Georg Thieme Verlag KG, 2016 (cit. on p. 19).

Ronaghi, Mostafa (2001). "Pyrosequencing sheds light on DNA sequencing." In: *Genome research* 11.1, pp. 3–11 (cit. on p. 74).

Sancar, Aziz, Laura A Lindsey-Boltz, Keziban Ünsal-Kaçmaz, and Stuart Linn (2004). "Molecular mechanisms of mammalian DNA repair and the DNA damage checkpoints." In: *Annual review of biochemistry* 73.1, pp. 39–85 (cit. on p. 33).

Sanger, Frederick, Steven Nicklen, and Alan R Coulson (1977). "DNA sequencing with chain-terminating inhibitors." In: *Proceedings of the national academy of sciences* 74.12, pp. 5463–5467 (cit. on p. 74).

Schena, Mark, Dari Shalon, Ronald W Davis, Patrick O Brown, et al. (1995). "Quantitative monitoring of gene expression patterns with a complementary DNA microarray." In: *SCIENCE-NEW YORK THEN WASHINGTON-*, pp. 467–467 (cit. on p. 75).

Schwarzenbach, Heidi, Dave SB Hoon, and Klaus Pantel (2011). "Cell-free nucleic acids as biomarkers in cancer patients." In: *Nature reviews. Cancer* 11.6, p. 426 (cit. on p. 59).

Shannon, Paul, 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." In: *Genome research* 13.11, pp. 2498–2504 (cit. on p. 77).

Shiny - Reactivity - An overview. URL: `https://shiny.rstudio.com/articles/reactivity-overview.html` (visited on 09/07/2017) (cit. on p. 63).

Shneiderman, Ben (1996). "The eyes have it: A task by data type taxonomy for information visualizations." In: *Visual Languages, 1996. Proceedings., IEEE Symposium on*. IEEE, pp. 336–343 (cit. on p. 14).

— (2010). *Designing the user interface: strategies for effective human-computer interaction*. Pearson Education India (cit. on p. 70).

Sim, GK, FC Kafatos, CW Jones, MD Koehler, A Efstratiadis, and T Maniatis (1979). "Use of a cDNA library for studies on evolution and developmental expression of the chorion multigene families." In: *Cell* 18.4, pp. 1303–1316 (cit. on p. 75).

Slack, Jonathan (2014). *Genes: A Very Short Introduction*. OUP Oxford (cit. on p. 31).

Southern, Edwin Mellor (1975). "Detection of specific sequences among DNA fragments separated by gel electrophoresis." In: *Journal of molecular biology* 98.3, pp. 503–508 (cit. on p. 36).

Srinivasprasad, Vijayan, Janardhanam Dineshshankar, J Sathiyajeeva, M Karthikeyan, J Sunitha, and Ramachandran Ragunathan (2015). "Liaison between micro-organisms and oral cancer." In: *Journal of pharmacy & bioallied sciences* 7.Suppl 2, S354 (cit. on p. 35).

Stewart, BWKP, Christopher P Wild, et al. (2014). *World cancer report 2014*. Tech. rep. WHO (cit. on p. 29).

Sutcliffe, J Gregor, Robert J Milner, Floyd E Bloom, and Richard A Lerner (1982). "Common 82-nucleotide sequence unique to brain RNA." In: *Proceedings of the National Academy of Sciences* 79.16, pp. 4942–4946 (cit. on p. 75).

Sutherland, D Robert, LORI ANDERSON, MICHAEL KEENEY, RAKASH NAYAR, and IAN Chin-Yee (1996). "The ISHAGE guidelines for CD34+ cell determination by flow cytometry." In: *Journal of hematotherapy* 5.3, pp. 213–226 (cit. on p. 47).

Tatusov, Roman L, Eugene V Koonin, and David J Lipman (1997). "A genomic perspective on protein families." In: *Science* 278.5338, pp. 631–637 (cit. on p. 86).

Tatusov, Roman L, Natalie D Fedorova, John D Jackson, Aviva R Jacobs, Boris Kiryutin, Eugene V Koonin, Dmitri M Krylov, Raja Mazumder, Sergei L Mekhedov, Anastasia N Nikolskaya, et al. (2003). "The COG database: an updated version includes eukaryotes." In: *BMC bioinformatics* 4.1, p. 41 (cit. on p. 86).

The Noun Project. URL: `http://thenounproject.com/` (visited on 10/07/2016) (cit. on p. 19).

The World's Data. URL: `http://innovate.reduxio.com/the-worlds-data` (visited on 06/15/2017) (cit. on p. 5).

Thomas, Patricia S (1980). "Hybridization of denatured RNA and small DNA fragments transferred to nitrocellulose." In: *Proceedings of the National Academy of Sciences* 77.9, pp. 5201–5205 (cit. on p. 36).

Toonders, Joris (2014). "Data is the new oil of the digital economy." In: *Wired. https://www. wired. com/insights/2014/07/data-new-oil-digital-economy/(accessed 19 October 2016)* (cit. on p. 5).

Trypsteen, Wim, Matthijs Vynck, Jan De Neve, Pawel Bonczkowski, Maja Kiselinova, Eva Malatinkova, Karen Vervisch, Olivier Thas, Linos Vandekerckhove, and Ward De Spiegelaere (2015). "ddpcRquant: threshold determination for single channel droplet digital PCR experiments." In: *Analytical and bioanalytical chemistry* 407.19, p. 5827 (cit. on p. 43).

Tufte, Edward R (1990). *Envisioning information*. Graphics press (cit. on p. 13).

Velculescu, Victor E, Lin Zhang, Bert Vogelstein, and Kenneth W Kinzler (1995). "Serial analysis of gene expression." In: *Science* 270.5235, p. 484 (cit. on p. 75).

Verne, Jules (1864). "Journey to the Center of the Earth." In: Pierre-Jules Hetzel. Chap. XXXI (cit. on p. v).

Ware, Colin (2010). *Visual thinking: For design*. Morgan Kaufmann (cit. on pp. 10, 13, 66).

Weinstein, John N, Eric A Collisson, Gordon B Mills, Kenna R Mills Shaw, Brad A Ozenberger, Kyle Ellrott, Ilya Shmulevich, Chris Sander, Joshua M Stuart, Cancer Genome Atlas Research Network, et al. (2013). "The cancer genome atlas pan-cancer analysis project." In: *Nature genetics* 45.10, pp. 1113–1120 (cit. on p. 70).

Wetterstrand, Kris A (2017). *DNA Sequencing Costs: Data from the NHGRI Genome Sequencing Program (GSP)*. URL: `www.genome.gov/sequencingcostsdata` (visited on 10/15/2017) (cit. on p. 75).

Zare, Habil, Parisa Shooshtari, Arvind Gupta, and Ryan R Brinkman (2010). "Data reduction for spectral

clustering to analyze high throughput flow cytometry data." In: *BMC bioinformatics* 11.1, p. 403 (cit. on p. 48).

Zhang, Lewei, Catherine F Poh, Michele Williams, Denise M Laronde, Ken Berean, Pamela J Gardner, Huijun Jiang, Lang Wu, J Jack Lee, and Miriam P Rosin (2012). "Loss of heterozygosity (LOH) profiles—validated risk predictors for progression to oral cancer." In: *Cancer prevention research* 5.9, pp. 1081–1089 (cit. on p. 35).

Zhang, Weiwen, Feng Li, and Lei Nie (2010). "Integrating multiple 'omics' analysis for microbial biology: application and methodologies." In: *Microbiology* 156.2, pp. 287–301 (cit. on pp. 73, 76).

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede and Ivo Pletikosić. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". `classicthesis` is available for both LaTeX and LyX:

`https://bitbucket.org/amiede/classicthesis/`

Happy users of `classicthesis` usually send a real postcard to the author, a collection of postcards received so far is featured here:

`http://postcards.miede.de/`

Thank you very much for your feedback and contribution.

Final Version as of July 14, 2018 (`classicthesis` v4.4).