Workflow tools for biological applications

by

Marie C Vendettuoli

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Co-majors: Bioinformatics and Computational Biology

Human Computer Interaction

Program of Study Committee:

Heike Hofmann, Co-major Professor

Dianne Cook, Co-major Professor

Christopher Harding

Anthony Townsend

Eve Wurtele

Iowa State University

Ames, Iowa

2013

DEDICATION

I dedicate this work to

Ted Hoagland, my best friend and soulmate.

my mom, Patricia Chan Vendettuoli, for always encouraging me to pursue my goals in academics and life.

the memory of my father Anthony Mark Vendettuoli, my grandmother Lee Toa and my uncle Dr. LaVaughn Hales, whose lives are a constant source of inspiration.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

With much thanks for the support of professors, mentors, collaborators and colleagues who helped me with various aspects of research and writing. My major advisor Dr. Heike Hofmann, who patiently balances sagacity with indefatigable creativity in every conversation. My major advisor Dr. Dianne Cook, whose teaching style is a positive enthusiasm I can only aspire to emulate. The members of my Program of Study committee: Dr. Anthony Townsend, Dr. Eve Wurtele, and Dr. Chris Harding, each of whom has been a direct and positive influence on my research and professional development.

ABSTRACT

When identifying best practices for multistep processes involving data analysis, it is frequently the case that the data scientist is asked to wear many hats simultaneouly: developer, programmer, statistician, graphic designer, writer, administrator. Although many scientists address these roles with great success, it is often at the expense of reproducibility, scalability, and organizational knowledge. The process of formalizing each step of the process creates opportunity to apply lessons learned and proven tools from multiple disciplines to optimize each step of the transformation from raw data to usable output. This modular approach allows organizations to mix off the shelf technical solutions with custom, swap out components for flexibility and minimize rework.

The primary focus of this dissertation is to extend the conceptualization of pipeline to include methods drawn from human computer interaction, exploratory data analysis, interactive graphics, and reproducible research. We describe application to three distinct user groups: (1) a general audience of readers (2) biologists involved in metabolomics analysis (3) analysts working in a public sector regulatory environment. The resulting technical tools are implemented in the R packages **ggparallel**, **chromatoplotsGUI**, **dataFormats**, and **CVBreports**.

Our analysis shows that these tools facilitate a positive transformative effect on the quality of communication between stakeholders. Specifically we see that the common angles plot presented in **ggparallel** reduces the lie factor, **chromatoplotsGUI** enables display of metabolomic data rapidly and with a level of detail that facilitates development of the underlying analysis engine and the methods of **dataFormats** and **CVBreports** enable significantly reduced turnaround times for preliminary data assesment.

CHAPTER 1. INTRODUCTION

1.1 Overview

Data scientists in both academia and public sector roles are faced with a similar challenge: analyze an increasing volume of data of greater complexity while adhering to (and improving) existing quality standards. Due to economic pressure, a successful solution is dictated in terms of accessibility (accuracy of results, ease of use, reduced cognitive load) and resource allocation as well as traditional metrics including speed, accuracy, reproducibility and ease of use.

The conceptual framework relating initial data, deliverables (e.g. graphical objects, reports, recommendations, other representations) and the process of transformation between these states is referenced:

In business literature as 'workflow': [van der Aalst and van Hee (2004); van der Aalst et al. (2003); Georgakopoulos et al. (1995)]

In computing literature as a 'pipeline': both in reference to general processing: [Patterson and Hennessy (2011); Koutroumpas and Higgins (2008); Kroening and Paul (2001); Grunbacher (1998)] and specifically in application to graphics rendering [Tenllado et al. (2008); Liu et al. (2008); Fernando and Pharr (2004)].

Within the scope of interactive graphics software: This pipeline model is extended to include bidirectional communication and commander management [Wickham et al. (2009); Sutherland et al. (2000)]

Motivating development for key tools in exploratory data analysis: DataViewer [Buja et al. (1991, 1987); Hurley (1987)] , XGobi [Swayne et al. (1998); Symanzik et al. (1996); Swayne et al. (1991)] and GGobi [Cook and Swayne (2007); Swayne et al. (2003)]

Most recently the suite of tools developed using `R` package *cranvas*: [Vendettuoli and Hofmann (2012); Cheng (2011); Wickham and Hofmann (2011); Yin (2011)]

It is in a context that extends the definition from interactive graphics software that the term 'pipeline' is used for the remainder of this proposal, while 'workfow' refers to a specific instance of implementation (e.g. a specific user group).

A related but separate sphere of research addresses concepts supporting best practices for implementing pipeline components, including: grammar of graphics [Wickham (2010a); Wilkinson (2005)], interactive visualization [Luse et al. (2008); Koehring et al. (2008); Nordvik and Harding (2008); Foley and Van Dam (1982)], human computer interaction [Harding and Souleyrette (2010); Carroll (2009); Rosson and Carroll (2001); MacKenzie (1992)] and reproducible research [Baggerly and Coombes (2009); Gentleman and Temple Lang (2004); Buckheit et al. (1995)]. Display design in the field of human computer interaction (HCI) four categories of principles: perception, mental model, attention, and memory [Wickens et al. (2004)] which emphasize a user centered design philosophy - systems must support and enhance natural inclinations of a user. Gestalt visual grouping principles identify practices for visualizations that present that information effectively. These include: proximity, similarity, continuity and common fate [Carroll (2009)]. Other concepts for good visualizations include those that facilitate interpretation, such as readable text (font sizing), grouping identifiers that are dissimilar (e.g. distinct colors), redundancy and mapping to physical models (when possible).

This dissertation outlines a framework for developing a workflows that extend interactive graphical pipelines to support workflows using tools from the fields of human computer interaction (including user perception of visualizations), and reproducible research. Figure 1.1 shows the role of each field in the overall framework, with example implementations presented in three case studies.

This paper is organized such that Chapter 1 introduces theory and logistics. Immediately after this paragraph, Chapter 1.2 provides greater detail regarding underlying concepts from previous literature. Chapter 1.3 describes work in progress: both material that this research

is dependent upon and gaps in existing literature that will be addressed by this research. Chapter 1.4 lists assumed knowledge competencies outside the scope of this paper and Chapter 1.5 summarizes the impact this dissertation has, both in the field and for a broader audience. Chapters 2, 3, and 4 includes text from submitted and proposed publications. Chapter 5 discusses tools that are available for distribution but not included in published work. Appendix A includes an additional, published, paper.

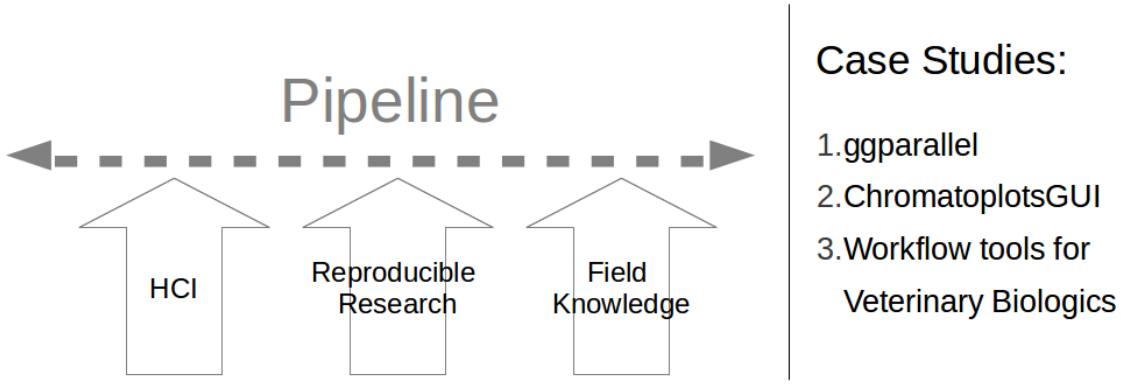


Figure 1.1: Relating case studies presented in this proposal to the underlying concepts. Pipelines are a conceptual approach to organizing implementation and analysis using principles from HCI, reproducible research and specific scientific disciplines to transform input data.

1.2 Background

Do analysts see themselves as part of the pipeline? Current perspectives envision the role of statistical analyst as a skilled craftsperson with experience in one or more applied settings [Davenport and Patil (2012); Fry (2004)]. This individual extends previous experiences to support design, organization and post-collection analysis of experimental data. The implication of this perspective is that in the course of performing their job, an analyst must demonstrate expertise in the tools used to manipulate, present and evaluate data sets. Centralizing all aspects of this process with one individual assumes that the scientist is highly skilled in each of these specialties equally [Cleveland (2001)], which may not be the case. This idiosyncratic approach reduces opportunity for reproducible research. It is also untenable once the size and/or volume of data exceed the capacity of an individuals working memory. From an organizational

perspective, a workflow driven by individuals is risky, as it links both quality and productivity to the tenure and ability of an individual. Additionally, that individual becomes a potential bottleneck for the entire workflow.

Defining systematic approaches to analysis via accessible tools is a means of knowledge transfer between scientists with similar roles [Zack et al. (2009)]. Designing pipelines that incorporate principles of human computer interaction is one way to ensure reproducibility and creates opportunities to leverage expertise from specialties such as computer programming, IT infrastructure development, and biological sciences. Furthermore, coupling design of representations with consideration of the target audience reduces potential for incorrect conclusions due to perceptual factors.

Pipelines describe a transformation process. Data is the initial starting material. Once data is modified, it becomes the starting material for the next stage. It is the analyst's interaction with visual representation(s) of the data that selects - from a set of predetermined operations - how the data is transformed and/or standardized. The logic a system employs to interpret user activity (gathered via use of mouse, keyboard or other haptic device) in making this selection may be referred as a set of rules by which a user communicates with the system. The set of possible operations (e.g. statistical summarization, color highlighting, subsetting) may also be referred to as the rules affecting content and direction of communication between stages of the pipeline. In a interactive graphical software pipeline, a single stage consists of - at minimum - the transformation or standardization rules, a representation allowing user interaction, and additional rules for capturing arguments via interaction [Wickham et al. (2009)].

Because rules in pipelines transmit information in both directions, one argument is that both human analyst and pipeline implementations (e.g. dataviewer, xgobi, ggobi, `R` packages: SixSigma, qcc, qualityTools) are elements of the same system. It is essential therefore to consider the quality of the information gained via interaction, which is directly affected both by the choice of representation and the analyst's understanding of rules governing interactive arguments.

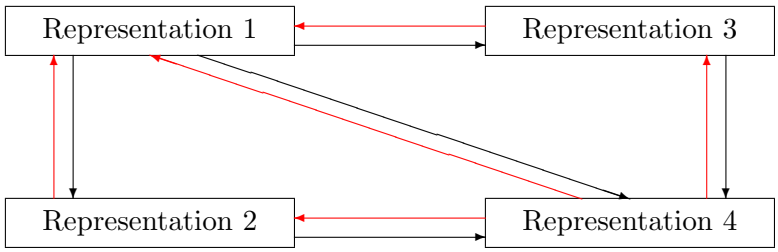
This research proposal focuses on exploring the boundary between the framework of data transformation and concepts underlying implementation practices.

Related work in literature may be divided into three categories:

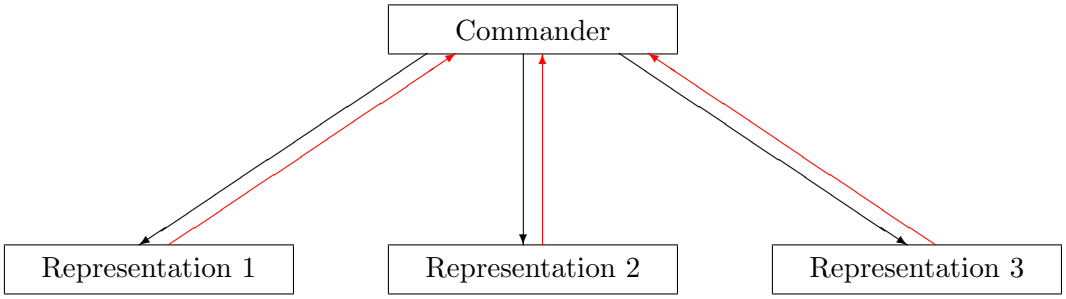
- Pipeline Concepts: How to organize and manage the flow of data from raw to deliverable state.
- Strategies for rule creation: Principles of perception and interaction from fields of graphics and human computer interaction
- Application: Field or specialty associated with a particular pipeline. Necessary to understand the impact and quality of that pipeline.

1.2.1 Pipeline Concepts

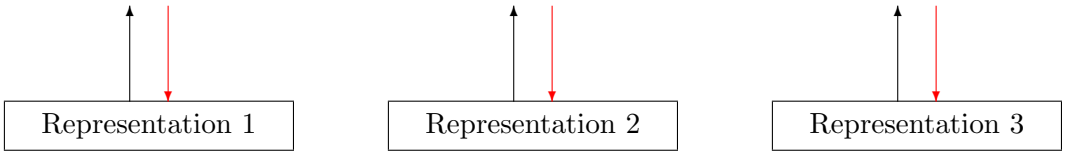
The first challenge within a pipeline is to define the general structure. Previous literature [Wickham et al. (2009)] suggests that the transformation portion of the pipeline may be contained, with a single data matrix as the input and a single data matrix as the output as shown in simplest case in Figure 1.2a. In this pipeline, communication channels must exist between the multiple visualizations so that the updates to data due to interaction with a single plot are propagated to other visualizations in a timely manner. Even when reducing the pipeline to modular components (see Figure 1.2c) or a more cumbersome approach using the commander model shown in Figure 1.2b, it is suggested that transformations may be encapsulated in a single stage [Wickham et al. (2009)]. However, the practical application of this philosophy faces a very real challenge of managing multiple transformations within a single pipeline, such as may occur when performing analysis on the same data set for multiple audiences. Figure 1.3 shows the extended framework. In this case, not only is there a need for bidirectional communication between stages of the pipeline, there may be a need for trigger events that update other transformations.



(a) Communication occurs between different data representations. This model is resource-intensive because each representation must be aware of all other representations.



(b) Managing pipelines using a commander. Communication to/from the commander becomes the bottleneck. Removing the commander destroys functional of the whole system.



(c) Modular communication strategy. Each representation is responsible for its own inbound and outbound communication.

Figure 1.2: Three models for handling communication within a pipeline (a) The nontrivial task of managing updating between multiple representation. (b) Using a commander to manage data updates due to user interaction. (c) Modular system where all data and updates are handled at communication between one stage and next, regardless of direction.

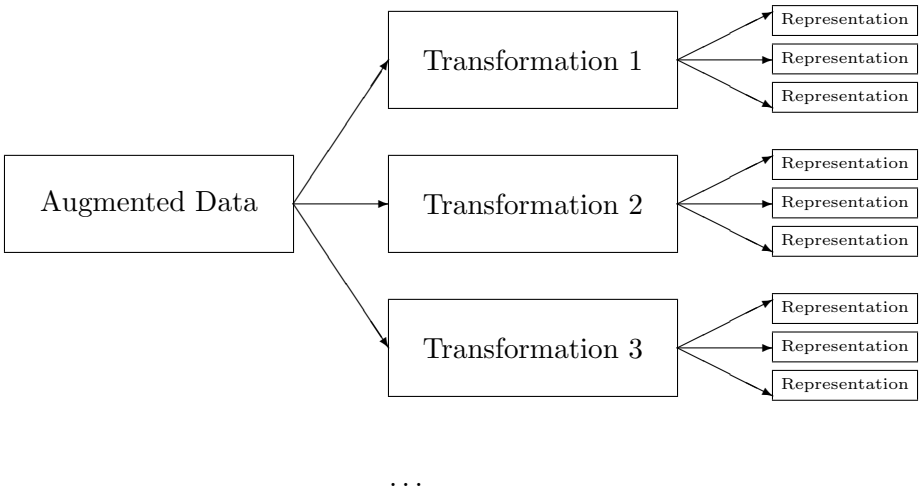


Figure 1.3: Pipeline framework extended for data undergoing multiple transformations. Implementing requires a significant increase in complexity compared to previous models.

A second challenge is how to store and access the data. Beyond the traditional data matrix - with columns representing variables - it is necessary to track variables that affect display in some manner. For a simple scatterplot, this may include display variables such as the glyph (i.e. shape), color or size of a data point. The choice of which values to augment the raw data matrix with affects the level of complexity that a visualization encodes. In the scatterplot example, including glyph *and* color allows for creation of visualizations that show two levels of groupings. Alternatively, including only one of the display variables would limit the possible groupings. Furthermore, the choice of display variables may have implications regarding the perception of implicit data structures. A basic case would be whether the variable encodes continuous or categorical information.

1.2.2 Rule Creation

If pipelines are a transformation process, rules define - step by step - *how* that transformation occurs. These rules govern every minutia of the pipeline: structure of the initial data matrix, each transformation of the data as it progresses along that pipeline in either direction, representation of data at each stage, interpretation of user interaction with a particular representation, and storage structure for any arguments or parameters obtained via interaction. Even the choice to not track values obtained thru interaction is a rule. The motivation to select a particular action (e.g. which data variables to include in a scatterplot, which statistical model to apply) to create an individual rule may come from a variety of fields including human computer interaction, reproducible research, and specific field knowledge for the application.

1.2.2.1 Strategies from Human computer interaction

When asking users to make interpretations using either single or multiple visualizations, it is also necessary to consider attention and memory models. Task-switching, even as simple as asking the user to look at a second visualization or wait while a new visualization is rendered, asks the user to store information in their working memory which reduces the amount of processing a user has accessible to address questions of greater complexity. Interaction (e.g. brushing of user-selected data points with a highlight color) provides one method to reduce cognitive load, but requires careful consideration in design - for example, interactions that rely on a mouse click should observe Fitts' Law and drop down menus must avoid presenting clutter while also limiting depth for ease of navigation.

Building interactive graphics software while keeping human computer interaction principles in mind presents an opportunity to both reduce incorrect communication between the system and user and transfer responsibility for routine processing to the system.

Interaction in exploratory analysis Exploratory data analysis by definition asks analysts to summarize without the influence of models or hypothesis, with visualization as a primary tool. It is a technique that aids in hypothesis *formation*, rather than hypothesis *testing*. Generating all possible displays that may apply to a data set may be a labor-intensive

endeavor - simply providing all scatterplots of n-dimensional data means $n!$ unique plots. Additionally, it may be that many of these plots are valuable to the analyst during this discovery phase but few used for later investigation or presentation. Interactivity provides a set of rules that reduces the user effort necessary to explore many visualizations. While the pipeline itself may reduce the number of visualizations that are applicable to the analysis, interactivity may not only further reduce the number of needed visualizations but also reduce the time and/or effort needed to link different mental models. In the simple scatterplot example from before, an interactive display may allow the analyst to highlight a grouping of data points and see how the pattern of that group changes when varying the variables of the axis (e.g. does the group stay tightly clustered? fragment into smaller groups?). Consistent visual groupings communicate a message of common fate under Gestalt principles.

Reducing the number of iterations required for sensemaking also reduces cognitive load in an exploratory data analysis (EDA) workflow. EDA is an iterative process by which statisticians use graphics to identify, refine and extend models that are targets for future confirmatory analysis [Tukey (1980)]. Employing graphics, both interactive and static, that are designed to reduce cognitive load at each iteration creates a multiplicative effect for overall cognitive demands. EDA is a powerful tool for examining data without the drawback of parametric or distributional assumptions, allowing researchers to simultaneously gain insight into both statistical significance and practical effect.

1.2.2.2 Strategies from Reproducible research

When presenting software solutions to computational problems, an ideal method of communication is to present readers with the tools necessary to reproduce the analysis and figures along with the descriptive text [Gentleman and Temple Lang (2004)]. This supports the philosophy of reproducible research, where a reader is presented with a complete environment that lets individuals interact with and explore analysis in a nonlinear manner [Buckheit et al. (1995)]. These encapsulated environments include data and specific instructions (code) for transforming raw data into the final product. An individual may simply be interested in recreating an author's work. Alternatively, they may seek to apply the same workflow to a different data set.

For academia and regulatory environments this process aligns well with organizational goals of scientific publications and audit requirements. In the former, there exists a common goal of reproducing existing work for pedagogical purposes and/or extending existing work into novel arenas. In the latter, stakeholders are interested in validating accuracy in documentation of procedures and decisions by checking for reproducibility.

1.2.2.3 Strategies from field knowledge

Metabolomics is the study of chemical entities (hormones, signaling molecules, metabolic intermediates, etc) that exist in a biological sample. The metabolomic profile (presence and relative abundance of each element) is dynamic and responsive to both environmental and genetic factors. To detect this profile, gas chromatography-mass spectometry (GC-MS) is a popular separation-detection technique, offering both high resolution and sensitivity.

Analysis in metabolomics is non-trivial in large part because the fragmentation pattern and/or identity of chemicals in the samples are unknown, unlike for traditional applications of mass spectrometry. Evaluating datasets with this additional unknown dimension greatly increases the complexity of the analysis workflow. Where a researcher would historically look at a specific m/z region for a known peak pattern (ratio of intensities [abundance] at specific m/z intervals) that is unique to that chemical fragment, they now must survey multiple spectra for consistent patterns. Complicating the identification phase of analysis is drift in m/z, intensity and/or retention time across replicate spectra in a single experiment. Different analysis tools implement distinctive algorithms for resolving this issue, but generally do not offer features for extending analysis beyond the identification stage. Once a list of unique metabolites is generated further analysis may take place, using information from biochemistry and findings by principal component analysis and/or random forests. For a complete analysis, researchers must apply tools in a modular manner.

Metabolomic analysis tools fall into one of three categories: proprietary, open-source and idiosyncratic. “Proprietary” software may be free, require costly licensing or anything in between. Of significance to the researcher are limitations imposed on both modification and accessibility to underlying code. These restrictions severely limit opportunities for reproducibility by

independent means or modifications that will improve the quality and/or performance of underlying algorithms. For scientists seeking to use proprietary software without modifications, the limited accessibility means results are generated without clear documentation regarding underlying algorithms a black box approach that ultimately detracts from result quality. Different proprietary software (and in some cases, versions of the same software) may bias results inconsistently or perpetuate assumptions that are incorrect for the experiment at hand.

With open-source systems, analysts can customize algorithms for specific application, or simply explore the code for greater understanding of implicit assumptions. Idiosyncratic methods include use of statistical software, business productivity tools, and manual solutions. Because a variety of software is involved, some of which has limited ability to track user actions, results are difficult to reproduce.

Business environments One business management strategy that has gained popularity through demonstrated success is Lean Six Sigma ("LSS") Lean manufacturing ("lean") is an approach to improving production workflow wherein activities that are not value-added for the customer are targeted for elimination. Six Sigma ("6σ") targets higher level business practices, asking for improvement strategies that are supported by data [Shah et al. (2008)]. To the business audience, these tools are marketed with a goal for reducing costs, improving quality and increasing revenue. When exploring the activities of implementing and enforcing a LSS environment, two features are apparent: (a) documentation of workflows and (b) emphasis on reproducible results.

1.3 Active Research & Opportunities for Further Development

chromatoplots [Lawrence et al. (2012a)] This package extends the available methods in package *xcms* [Tautenhahn et al. (2008); Smith et al. (2006)] for analysis of metabolomic data in `R`. *chromatoplots*applies the pipeline concept by first sorting the available functions into multiple workflow stages: generate profile, remove baseline, find peaks, find components, group components, retention time correction, summarize and normalize. Secondly, functions in the *chromatoplots* package use `R` classes and generic function definitions to track variables and

arguments necessary for bidirectional transformation. However, the package functions do not include augmented data variables to capture interaction between user and visual representations of data outside of the command line environment.

1.3.1 Information Gaps

In the exploration of existing practices for each of the cases previously identified, one opportunity for scientific development that emerges is the need for a framework to incorporate concepts from the diverse specialties described earlier: interactive graphics software, human computer interaction, reproducible research. Simply creating a tool for one specific audience using currently available technology limits effectiveness and quickly becomes outdated. Likewise, relying on the developer's current expertise in multiple fields ensures that, over time, the purpose and effectiveness of tools created are variable and inconsistent as new perspectives (and different developers) are brought on board. This dissertation identifies an accessible framework for developing tools that incorporate scientific disciplines of interactive visualization, reproducible research and human computer interaction in the course of exploring three distinct case studies.

Case 1: ggparallel It is often the case that the motivation to select a particular tool or method for task completion is drawn from techniques taught in a formal academic environment or observed informally (i.e. co-workers, classmates). In the case of applying parallel coordinates for categorical data values, the Parallel Sets (parsets) method has received much attention in both public media and academic literature [Blastland (2009); Bendix et al. (2005); Kosara (2012); Kosara et al. (2006); Davies (2012)]. Despite the popularity, the performance of this method as a tool for accurate communication of information was unclear: there is no documentation of robust usability testing in the literature.

The first part of this project is the technical implementation of two existing methods of presenting categorical data using parallel coordinates: parsets [Bendix et al. (2005); Kosara (2008, 2012); Kosara et al. (2006)] and hammock plots [Schonlau (2003)] as functions for the `R` Language Environment. A second component was creation and technical implementation of

alternatives to parsets and hammock plots. Two alternatives were developed. Common angles, which enforces the same angular orientation for all bi-variate connections (ribbons). Adjusted angle introduces a conversion factor so that not only are the ribbons oriented to the same angle, they are perceived to have the same thickness.

In addition to the implementation phase (package *ggparallel* [Hofmann and Vendettuoli (2012)]), a user testing phase of the project included comparisons between the four plot types (see: http://vrac.us2.qualtrics.com/SE/?SID=SV_a35sfAgwuhewc6x). Results from this survey are further explored in Section 2.

Case 2: chromatoplotsGUI Analysis tools in R packages: One weakness in tools for biological analyses in the field of metabolomics is the absence of options that leverage robust visualizations (i.e. incorporating concepts from human computer interaction within a reproducible environment (e.g. how to capture user settings in Chemstation software?). *chromatoplotsGUI* is a set of interactive visualization of each stage within the chromatoplots workflow, organized into a graphical user interface. While the package *chromatoplots* captures the analysis values in the pipeline framework and records the necessary values to recreate the analysis steps, it does not provide opportunity to extend pipeline management to information captured via an interactive graphic.

An additional step needed to finalize development of *chromatoplots* is validation using a known data set. For the chromatoplotsGUI interface, there are two sections: (1) creating functions to independently generate interactive graphics and (2) compilation of these interactive graphics into a single graphical user interface. The text for validation is in Chapter 4 and the code for *chromatoplotsGUI* may be found online at: https://github.com/mariev/chromatoplotsgui

Case 3: Statistical analysis for veterinary biologics Tools targeting the workflow (Figure 1.4) of statisticians at the Center for Veterinary Biologics (CVB) are much wider in scope than those in previous cases, in large part because it involves stakeholders from a variety of perspectives. One set of stakeholders include the firms submitting data. This audience is

presented with documentation and templates that ease the submission of data in a consistent structure. It is expected that, in most cases, submitting firms have limited knowledge of statistical analysis or data management and the tools must facilitate functionality in spite of this skill gap. Once the data is structured consistently, there is the challenge of how to document statistical analyses and what representations are the best choice for communicating findings to audiences of varying expertise and divergent perspectives. Differences between individuals' innate comprehension of tools and paradigms of related fields (e.g human computer interaction, reproducible research, interactive graphics) means that although work product from multiple statisticians superficially fills the same role when communicating to audiences, the quality and focus of that message differed greatly.

Formatting tools: Files that allow external firms to format data accurately, consistently and completely with minimal input from statisticians. Additionally, these files *must be accessible to firms without access to statistical software.*

Transformation tools: Functions written in `R` that transform the data into representations that incorporate concepts from human computer interaction and reproducible research.

Formalized development process: A development workflow that supports the framework outlined in this research. Active use of user testing to validate perceptual accuracy of visualizations.

Qualitative study: Interviews and performance metrics (e.g. turnaround times, number of firm resubmits) of the development process and tools.

Formatting guidance may be found online at: `http://www.aphis.usda.gov/animal_health/vet_biologics/vb_data_formats.shtml` - these documents have been subject to an internal USDA review process. The draft text summarizing research is included in Section 3. Package *PF* (for statistical summaries associated with prevented fraction data) is in distribution on `CRAN`.

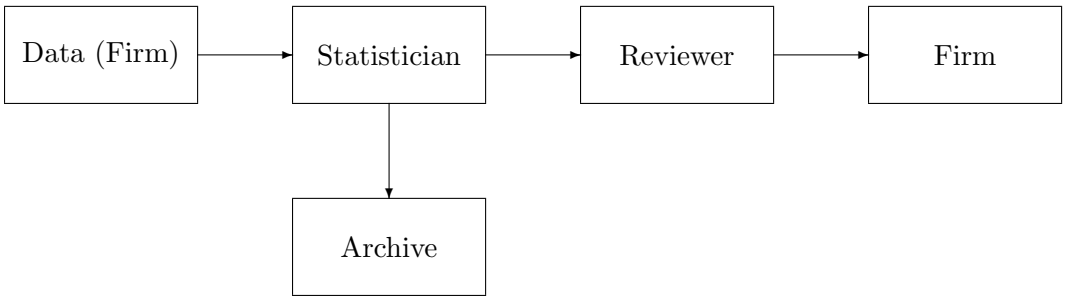


Figure 1.4: Overview of CVB Statistics workflow. Data is submitted by external firms, statisticians in CVB perform analysis. Reports are archived and made available to regulatory reviewers. Reviewer performs additional analysis and communicates findings to firm

1.4 Methodology

1.4.1 Assumptions

Recommendations that result from the findings of this research assume the organizational unit *as a whole* has access to the following competencies:

Software: `R` [R Core Team (2012); Hornik and Leisch (2002)] and add-on packages [Hornik (2012b,a); Fox (2009)], Excel [Walkenbach (2010b,a)], MikTeX/TeXLive or other LaTeXdistribution, Qt Libraries [Qt Project (2011)]

Languages: Passing familiarity and/or willingness to acquire basic programming skills in: `HTML`, LaTeX, `R`[1], tools for embedding R code in LaTeXdocuments (e.g. `Sweave` [Leisch (2002)], `knitr` [Xie (2012)] and/or `Markdown` [MacFarlane (2012a,b)]), `VBA`, command-line interaction

Infrastructure: Functional competence and access to subversion Apache (2012) or other version control system (e.g. [Git (2012)])

Culture: An environment of collaboration, iterative improvement, short development cycles, willingness to acquire new skills, dedication to quality improvement [Denison and Mishra (1995); Lencioni (2002); Robbins and Judge (2010)].

[1] `R` is considered both a statistical software environment and language

Each item on this list is associated with a technical or academic specialty of both greater depth and breadth than is the scope of this research. A limited review of relevent concepts is provided with each paper (Chapters 2, 3, 4).

1.4.2 Techniques

The topics in this section describe techniques employed both in order to conduct research and tools that are elements of the expected findings.

Usability testing To support user-centered design multiple usability tests will be conducted. A usability test consists of asking participants to complete a specific task in a structured environment. During the task, metrics are recorded, including: accuracy of task performance, duration spent performing a task (or sub task), and user preferences. A web-based survey is the simplest case, while asking participants to perform activities in a lab setting that is equipped to record actions (e.g. eye tracking, verbal and non verbal responses) is a more involved user test. Both types require approval by the institutional review board (IRB).

R Graphics R has demonstrated success in both academic and commercial endeavors, allowing users to rapidly perform statistical analysis on large volumes of data. Many packages developed by the user community undergo academic peer review regarding the soundness of both theory and implementation. In terms of implementation, R features an extensible open source programming environment and a strong infrastructure (documentation, online support). One area in R that has seen limited development since inception are graphics, meaning that base functionality does not capitalize on hardware improvements - specifically graphics cards - over the last two decades. Multiple packages exist as a workaround (qtbase/qtpaint, rgl, javagd) to access functionality provided by Qt, OpenGL, and Java libraries. Although these libraries represent a vast reduction of effort when compared to writing for C, implementing these packages requires programmers to understand the underlying programming constructs unique to each library environment. To do so represents a significant learning curve for those new to programming. Even for individuals experienced programming in R or other languages,

there is a learning curve associated with acquiring fluency with new syntax, methods and classes. For researchers and statisticians at the initial stages of identifying what graphical tools best fit their goals, this obstacle may prevent even the attempt to acquire skills essential for a successful implementation. Regardless of user skills, the absence of off-the-shelf visualization solutions that support human computer interaction paradigms limits user ability to generate quality graphics and interaction during the skill acquisition process.

Interactivity in `R` Graphics that are natively available in `R` are static. Add-on packages: `rgl, qtbase` allow access to interactive graphics without exiting the `R` workspace. Additionally, packages such as `tcltk, gWidgets and qtbase` allow for inclusion for navigation tools (e.g. drop down menus, clickable buttons) with an appearance that is consistent with the user's computing environment (i.e. Windows, Mac, or Linux). The challenge with packages listed that allow for interactive graphics is that they primarily provide access to low-level plotting elements (e.g. line, dot, glyph) while base graphics provide access to high-level functions (e.g. axis, boxplot, barplot). The new package *cranvas* [Xie et al. (2012)] is a package that provides high-level access to interactive graphic. It may be found at: https://github.com/ggobi/cranvas.

Reproducible reporting in R: Sweave and knitr Two `R` tools that support reproducible reporting are Sweave and knitr. The former is a system that integrates chunks of LaTeXwith R code, using files of the extension `.rnw`. The benefit is that a single `.rnw` file contains all text *and* the `R` code involved in analysis and figure generation. *knitr* extends this framework with output to multiple languages, including markdown, html and restructured text.

Publishing and collaboration via svn and github As those with experience creating documents in a collaborative environment can attest to, shared access to a common file structure is insufficient. While it is beneficial to see the latest changes a coauthor has added within minutes of real-time, it can be cumbersome to coordinate the timing of each contributors efforts. The negative consequences are primarily the potential for lost work. One software solution to address this is subversion Apache (2012). Another approach are git tools such as the popular github [Git (2012)]. This last option offers the greatest transparency and widest

distribution platform and it is what will be used to distribute code discussed in this proposal while it is in a development state.

1.5 Significant contributions of dissertation

The framework presented in this dissertation has origins in reducing the amount of rework necessary for workflow development when implementing the technical solutions presented in each of three case studies. In the process of creating and implementing pipelines useful for each of these highly specialized audiences, it became apparent that existing models for development neglect basic concepts from fields of human computer interaction, interactive graphical software and reproducible research. Furthermore, this neglect transcends the size of the affected community or the perceived quality of the scientists involved. At one level, the case studies with their diversity of of target audiences are demonstrations of the broader impact of this research. Additionally:

1.5.1 ggparallel

The research in this paper addresses the issue of missing user testing for the popular visualization method parsets. Although the parsets approach has been written up both in academic and mainstream media, testing for the effectiveness of parsets's ability to communicate the underlying data has not been previous presented.

Technical implementation of *ggparallel* is in distribution on `CRAN` and at `https://github.com/heike/ggparallel`. User testing was conducted in survey form under IRB approval and may be found at: `http://vrac.us2.qualtrics.com/SE/?SID=SV_a35sfAgwuhewc6x`. The research paper associated with this case study was accepted by *IEEE Transactions on Visualization and Computer Graphics* June 2013, is reproduced in Chapter 2 and includes the following components:

1. Survey of existing options for displaying parallel coordinates of categorical data.
2. Implementation of parallel set and hammock plot algorithms in package *ggparallel*.
3. Propose a new display, common angle plots and implement in package *ggparallel*.

4. Perform usability testing to evaluate performance of this new display against existing tools.

5. Propose a quantitative measurement of the line-width illusion, a lie factor effect which explains reduced user performance when reading information encoded in hammock plots and parallel set charts.

1.5.2 chromatoplotsGUI

The motivation for development of chromatoplotsGUI is to address the need for analysis of metabolomic data in a reproducible and extensible environment that involves interactive visualizations. Commercial tools (e.g. Chemstation) may embed interactive graphics, but limit the user to analytic algorithms specified by the developer. Open source tools (e.g. xcms) may allow for extensibility, but offer limited options (none of which are interactive) for displaying data graphically.

Underlying infrastructure for the interactive graphics of *chromatoplotsGUI* is the new `R` package *cranvas* [Xie et al. (2012)]. At this time, there are only trivial examples of using *cranvas* for exploratory data analysis. *chromatoplotsGUI* may act as a teaching example. Package *chromatoplots* is newly developed and the development of the GUI supports its validation.

Chapter 4 discusses testing to validate the chromatoplots engine and preliminary coding for visualizations and interface may be found at: `https://github.com/mariev/chromatoplotsgui`. Release of *chromatoplotsGUI* will follow outlet selected for *chromatoplots*, which has yet to be determined. While *chromatoplotsGUI* can be used independently of chromatoplots, with xcms as the analysis engine, greater flexibility regarding the curve fits matching across spectra is enabled. The project consists of the following elements:

1. Validate *chromatoplots* using data generated from samples with known content.

2. Create interactive visualizations in R for all stages of processing in *chromatoplots*, individually

3. Create a graphical user interface that incorporates visualizations created into a single framework that illustrates concepts underlying metabolomics analysis

1.5.3 Veterinary Biologics

While the tools developed are for daily use by statisticians at CVB, expectations of formatting data submissions affect the workflow of over 300 firms in industry. For smaller firms and operators without access to a mature data management team, the published guidance is an educational resource.

The paper summarizing this research may be found in Chapter 3. The first `R` package, *PF* is in distribution on `CRAN`. Additional packages *dataFormats* and *CVBreports* may be found at https://github.com/mariev. Documentation associated with the data formatting standards are available at: http://www.aphis.usda.gov/animal_health/vet_biologics/vb_data_formats.shtml.

1. Develop data format standards. Document and communicate to internal and external stakeholders via hand-on training, email consultations, publicly accessible web pages and help pages. Use of these data formating standards have reduced the amount of rejected data submissions from approximately 50% at project start to current rates of <5%.

2. Create R packages to transform raw data (in specified data formats) into visualizations and summaries useful for preliminary data analysis. Package *PF* is currently available on CRAN. Packages *dataFormats* and *CVBreports* are distributed via github.

3. Create LaTeXtemplates for the processing of routine reports to meet business objectives.

4. Perform a qualitative study to evaluate impact of these tools in the CVB Statistics group. Turnaround time for initial processing of data submissions is now on the order of 48 hours versus historical turnaround times of up to 4 months.

CHAPTER 2. COMMON ANGLE PLOTS AS PERCEPTION-TRUE VISUALIZATIONS OF CATEGORICAL ASSOCIATIONS

A paper accepted by IEEE Transactions on Visualization and Computer Graphics

This paper presents the underlying research that supports development of the `ggparallel` package, published to CRAN in Fall 2012. It is a collaborative effort with H Hofmann. The paper, included below, was accepted June 2013 for InfoVis 2013, with publication in a special issue of IEEE Transactions on Visualization and Computer Graphics (provisional acceptance rate 25%)

Abstract

Visualizations are great tools of communications - they summarize findings and quickly convey main messages to our audience. As designers of charts we have to make sure that information is shown with a minimum of distortion. We have to also consider illusions and other perceptual limitations of our audience. In this paper we discuss the effect and strength of the line width illusion, a Müller-Lyer type illusion, on designs related to displaying associations between categorical variables. Parallel sets and hammock plots are both affected by line width illusions. We introduce the common-angle plot as an alternative method for displaying categorical data in a manner that minimizes the effect from perceptual illusions. Finally, we present results from user studies as evidence that common angle charts resolve problems with the line width illusion.

2.1 Introduction

A well-designed graph is a powerful tool that transcends barriers of language to communicate complex concepts from author to audience. Problems arise when readers are unable to easily extract a chart's main message or are led to wrong conclusions due to distortions. This endangers the trust between readers and creators of charts, which is based on the main premise that graphics have to be true to the data Tufte (1991); Wainer (2000); Robbins (2005). There is a lot of discussion on keeping true to the data in the framework of (ab)using three dimensional effects in graphics. Tufte Tufte (1991) goes as far as defining a *lie-factor* – the ratio of the size of an effect in the data compared to the size of an effect shown. Any large deviation of this factor from one indicates a misuse of graphical techniques. Computational tools help us ensure technical trueness – but this brings up the additional question of how we deal with situations that involve innate inability or trigger learned misperceptions. One example of distortions of this kind is the Müller-Lyer family of illusions, which include contextual illusions, such as differently perceived lengths of line segments depending on the orientation of arrow heads or the sine illusion Day and Stecher (1991).

Regardless of the cause of distortion, it is the responsibility of the author of a chart to create visualizations that allows readers to extract an accurate interpretation of the underlying data. In order to gauge the extent of distortion due to perceptual limitations, we can employ user studies to provide empirical evidence supporting underlying cognitive models or previously unknown or not anticipated illusions.

Parallel sets (parsets) Kosara et al. (2006) are a graphical method for visualizing multivariate categorical data. Since their initial publication, parallel sets have spread to mass media outlets Kosara (2012); Carter and Bostock (2012); Blastland (2009), have been implemented in various languages Kosara (2012); Bostock et al. (2011); Davies (2012) and are a reputable resource for further academic work (Kosara et al. (2006) has 70 citations per Google scholar). While retaining the ability to visualize a large number of dimensions simultaneously that is the parallel coordinates' hallmark trade, parallel sets combine with it a frequency scale that is a well-known feature of other categorical displays such as barcharts or mosaic plots Hartigan

	Crew	1st	2nd	3rd
Survivors	212	203	118	178
Non-Survivors	673	122	167	528

Table 2.1: Correct ordering of variable Class is: crew, first class, third class, followed by second class.

and Kleiner (1982); Friendly (1992); Hofmann (2000); Theus et al. (1997). Unfortunately, the parallel set plot is a victim to distortion due to a contextual illusion: consider the parset plot of Figure 2.1. This plot shows the relationship between class status and survival on board the

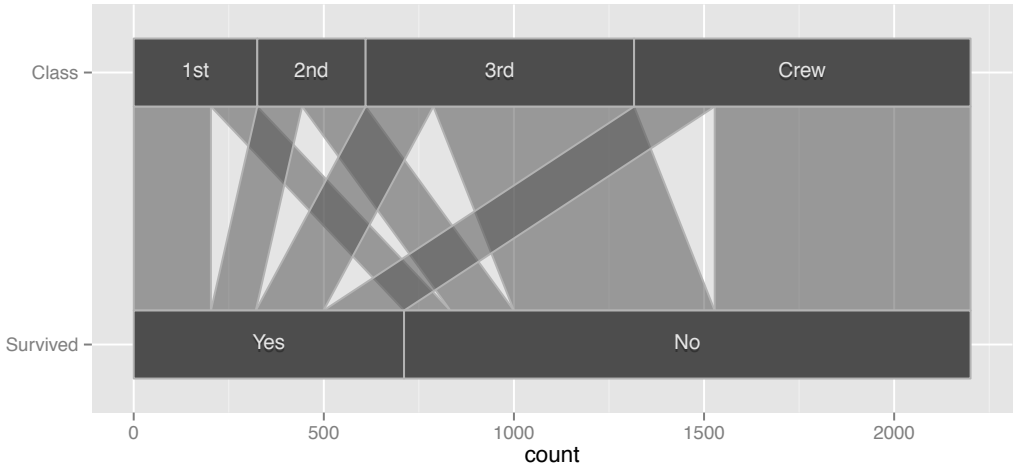


Figure 2.1: Parallel sets plot showing the relationship between survival of the sinking of the HMS Titanic and class membership.

HMS Titanic Dawson (1995). Class status is recorded as either crew member or passengers in first, second, or third class. The top bar in figure 2.1 shows the variable Class. The bottom bar shows survival as yes and no. Lines are drawn between top and bottom bar – the (horizontal) width is proportional to the number of survivor and non-survivor they represent. A reasonable task based on this chart is to order levels of the variable 'class' by number of survivors. However, when study participants were asked to perform this task, only 12.5% respondents selected the correct order, see table 2.3.

We believe that readers view parallel sets they are subject to the *line width illusion*, a perceptual distortion that we describe and quantify in this paper. We also propose and test

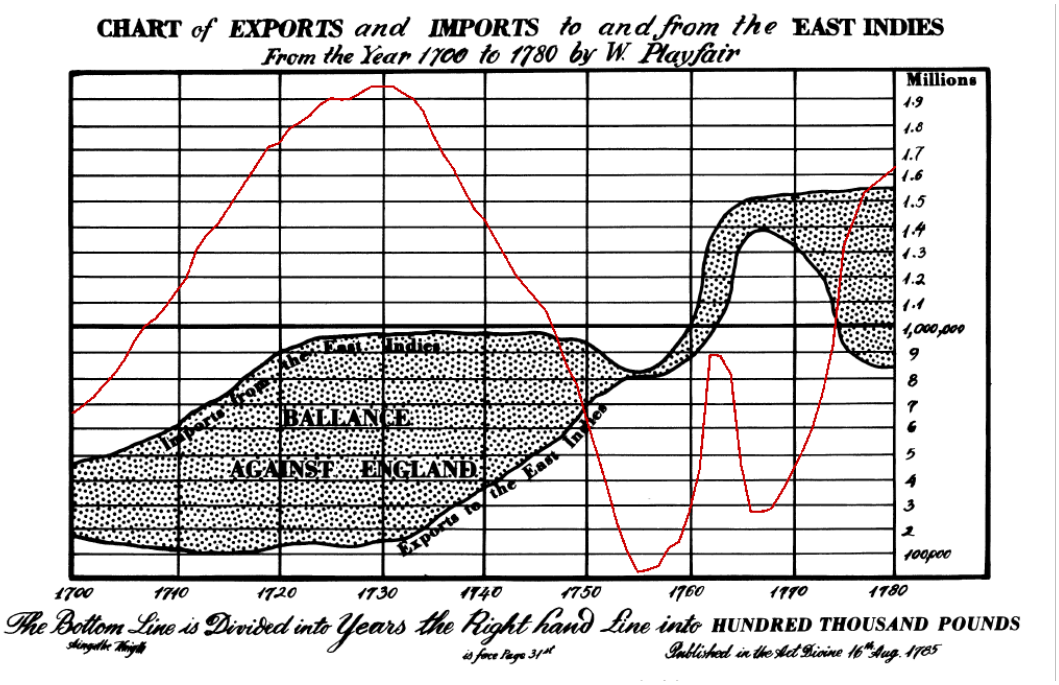


Figure 2.2: Playfair's chart from the Commercial and Political Atlas (1786) showing the balance of trade between England and the East Indies. In which years was the difference between imports and exports the highest?

common angle plots, an alternative graphing method for visualizing multivariate categorical is not subject to the *line width illusion*.

2.2 Line width illusion

An example of the *line width illusion* is displayed in figure 2.2. This chart displays the balance of trade between England and the East Indies as shown by William Playfair in his Commercial and Political Atlas, 1786 Playfair (1786); Playfair et al. (2005). One purpose of this chart is to demonstrate the difference between imports and exports in a particular year and its pattern over that time frame. The difference in exports and imports is encoded as the vertical difference between the lines. When observers are asked to sketch out the difference between exports and imports Cleveland and McGill (1984), they very often miss the steep rise in the difference between the lines in the years between about 1755 and 1765. Figure 2.3 shows the actual difference between imports and exports.

This phenomenon is known and widely discussed in statistical graphics literature Cleveland

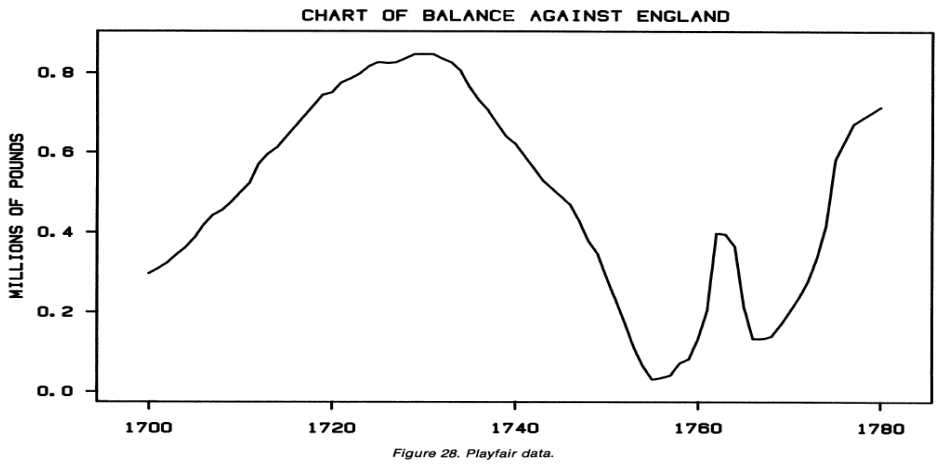


Figure 2.3: Difference between exports and imports from England to and from the East Indies in the 18th century – the steep rise in the difference around 1760 comes as a surprise to many viewers of the raw data in figure 2.2.

and McGill (1984); Tufte (1991); Wainer (2000); Robbins (2005). It is due to our tendency to assess distance between curves as the minimal (orthogonal) distance rather than the vertical distance – see sketch 2.4 for a visual representation of both.

In the perception literature, this phenomenon is known as part of a group of geometrical optical misperceptions of a context-sensitive nature classified as Müller-Lyer illusions Day and Stecher (1991). Interestingly, there seems to be a general agreement that this illusion exists, but a quantification of it is curiously absent from literature.

The type of chart as shown in figure 2.2 proposed by Playfair is shown quite commonly, particular in election years – where these kind of charts are used to enable comparisons of support for several candidates, the recommendation from literature is to avoid charts in which the audience is asked to do visual subtractions, and show these differences directly.

2.2.1 Strength of line width illusion

When visually evaluating lines of thickness greater than one, the line width illusion applies, only now the *edges* of a single line take on the role of the separate curves. As above, there is a strong preference of evaluating the width of lines orthogonal to their slopes as opposed to horizontally (see figure 2.4) needed for a correct evaluation of parallel sets-style displays.

Orthogonal w_o and horizontal w_h line widths are related – the orthogonal line width depends

on the angle (or, equivalently, the slope) of the line:

$$w_o = w_h \sin\theta, \tag{2.1}$$

where θ is the angle of the line with respect to the horizontal line.

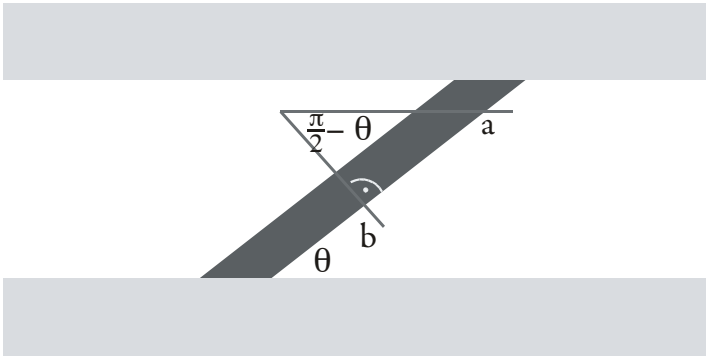


Figure 2.4: Sketch of line width assessments: (a) is showing horizontal width, (b) shows width orthogonal to the slope. Survey results in section 2.5.2 indicate that observers associate line width more with orthogonal width (b) than horizontal width (a).

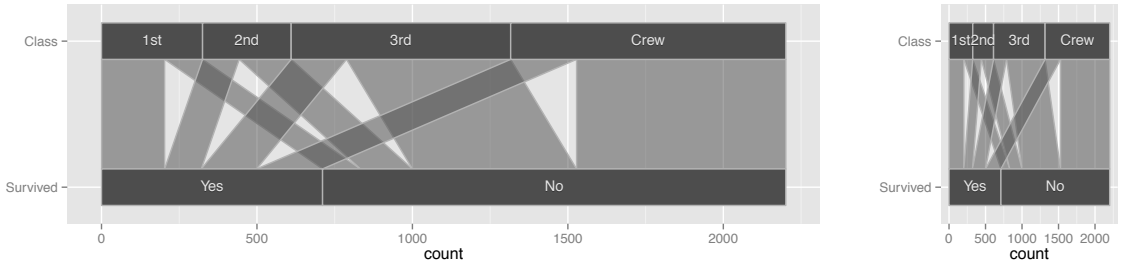


Figure 2.5: Parallel sets plots of survival on the Titanic by class. Different aspect ratios seemingly change the thickness of line segments, compare e.g. number of survivors in 3rd class and in the crew.

The perceived slope of a line very much depends on the aspect ratio of the corresponding plot – changing the height to width ratio of a display will change our perception of the corresponding line widths, if they are not adjusted for the slope Cleveland and McGill (1984). This finding is not new, but its strength on our perception is surprising, as can be seen in the example of figure 2.5. Again, survival and class membership on the Titanic is shown; the same parallel

sets plot is shown twice in this figure, but with very different aspect ratios: in the plot on the left the number of surviving 3rd class passengers seems to be about twice as big as the number of survivors among crew members, whereas in the plot on the right the lines have about equal (orthogonal) width. Obviously, this is not due to a change in numbers.

For parallel sets-style displays, the audience has *area of the line segment* an alternate visual cue when evaluating frequencies. Because height (or width for a rotated display) of line segments is constant across the display, the width of a particular segment is proportional to its area. We can therefore employ area comparisons as a proxy or to augment line width evaluations. However, existing literature suggests that this method of comparison is particularly prone to errors in two scenarios commonly seen in parallel sets: (1) extreme aspect ratios of the rectangular shape Heer and Bostock (2010) and (2) when comparing rectangles rotated relative to each other Kong et al. (2010). This incorrect perception and comparison of areas distorts the message readers discern from the graph.

2.3 Related work

Hammock plots, introduced by M Schonlau in Schonlau (2003), provide an alternative to parallel sets that is adjusted for the line width illusion. This is done by adjusting the –horizontal– line width by a factor of $\sin\theta$, as discussed in equation (2.1). This adjustment makes the perceived –orthogonal– line width to be proportional to the number of observations it represents. Figure 2.6 shows an example of a four dimensional hammock plot of the Titanic data. From top to bottom Class, Gender, Survival, and again Class are shown.

Similarly to the parallel sets plot, the bars are divided according to class membership numbers. Lines connect categories between neighboring variables, orthogonal line widths are representing the number of individuals in each combination. Unlike the parallel sets, the lines start from the middle of the bin and connect to the middle of the other variable's bins. This convention is in part due to the fact that the sum of horizontal widths (w_h) after adjustment is greater than the width of marginal bars.

The graph shows that barely any women were in the crew, while male crew members make up the second largest contingent overall. While overall a few more men survived than women,

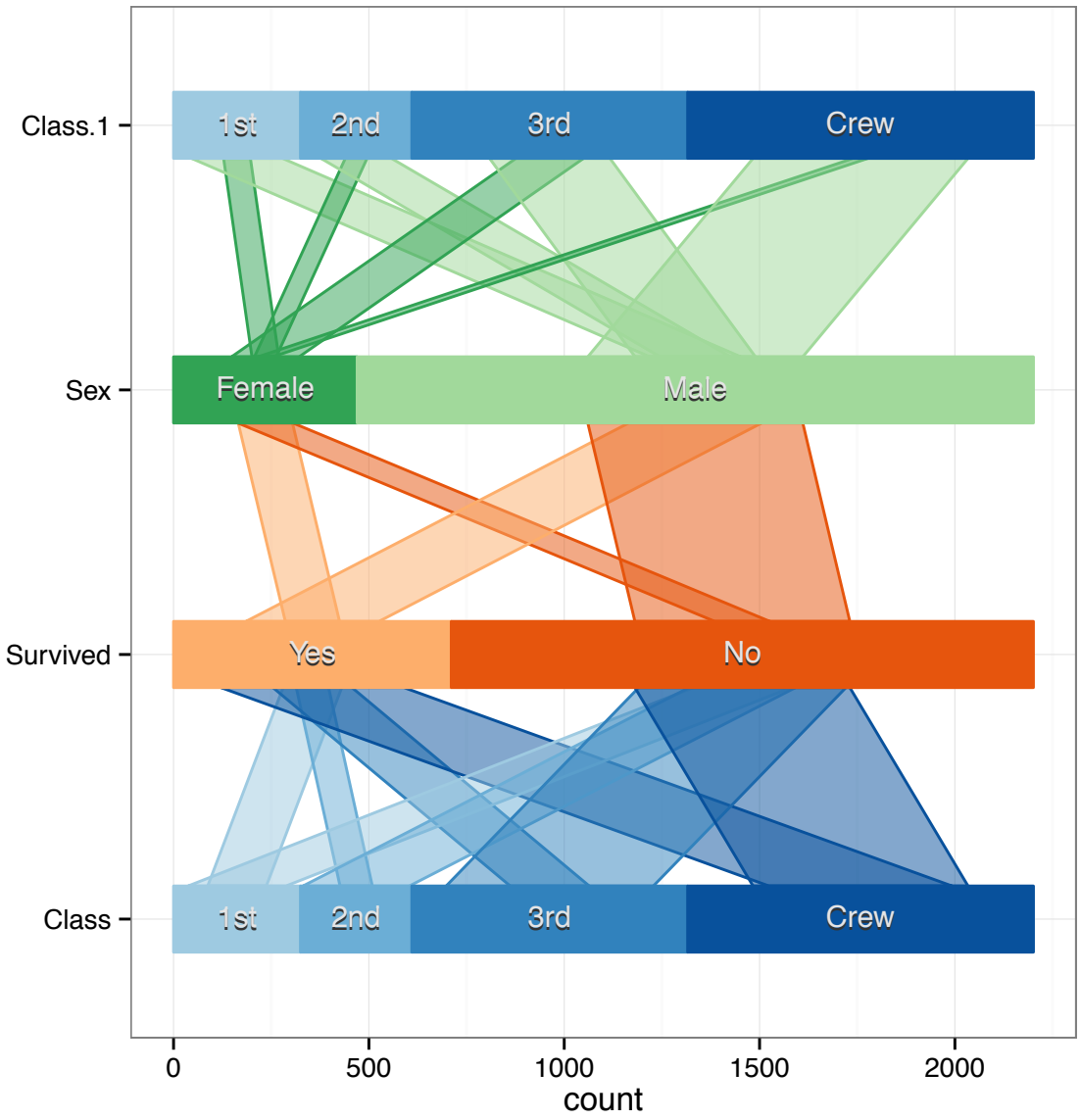


Figure 2.6: Hammock plot of the relationship between Class and Survival on the Titanic.

proportionally the situation is much different – a much higher percentage of women survived than men. While more first class passengers survived than not, the survival chances of second class passengers were evenly divided. For third class passengers and crew members fewer members did survive than not.

As the adjustment of line widths is made with respect to the angle θ, which itself depends on the aspect ratio of a plot, we need complete control over these properties of the plotting device when constructing hammock plots – in our implementation (see below for details) we have dealt with this issue by fixing the aspect ratio. This is problematic in some situations, where the rendering has to be done without knowledge of the plotting device.

2.3.1 Reverse linewidth

A problem that arises in evaluating hammock plots is that if an observer focuses on horizontal line width the plots suffer from a *reverse line width illusion*: judging the number of survivors by class in figures 2.6 and 2.7 based on horizontal line width results in an ordering of (largest to smallest) Crew, 3rd, 1st, and 2nd – which is not correct either. Because the lines are centered around the middle of each level, a contextual coordinate system is imposed that encourages comparisons of horizontal width. However, horizontal width is no longer proportional to underlying data, because of the line width adjustment. Rearranging equation 2.1,

$$w_h = w_o \csc\theta, \tag{2.2}$$

where w_o is proportional to observations and θ is the angle of the line with respect to the horizontal line.

Further supporting the poor context is the issue that the bands (unlike parallel sets) are no long proportional in area to the underlying data. Previous work Heer and Bostock (2010); Kong et al. (2010) has shown that audiences experience perceptual distortion when comparing (1) extreme aspect ratios of the rectangular shape and (2) when comparing rectangles rotated relative to each other, both of which apply to hammock plots.

Figure 2.7: Lines in hammock plot of Titanic data for survival variable, level yes. Comparing horizontal widths suggests that a greater number of survivors were from third class instead of first, which is inconsistent with underlying data.

2.4 Common angles

Figure 2.8 shows a common angle plot of the same data as the hammock plot.

As in the previously discussed display types ribbons are drawn between categories with widths that are proportional to the number of records they represent.

In order to ensure that widths of all bands are comparable without any distortion, their slopes are artificially made the same in the following manner: assuming a vertical display as shown in figure 2.8, connecting bands between categories are a combination of a vertical segment, a segment under a pre-specified angle θ, followed by another vertical segment. The pre-specified angle θ (between the line and the vertical band) is given as –at most– the angle of the longest connecting line between two categories of neighboring variables. This makes the width of ribbons comparable without being affected by the distortion, as all ribbons are sharing at least one segment under the same angle.

Common angles, plus the related methods of hammock plots and parallel sets are implemented in the package `ggparallel` based on the `ggplot2` Wickham (2009) plotting framework in the software R 2.15.1 R Core Team (2012). The ggparallel package is freely available from CRAN (http://www. r-project.org/). The colors for the plots have been chosen using color schemes from the ColorBrewer project Harrower and Brewer (2003), as implemented in the R package RColorBrewer Neuwirth (2011).

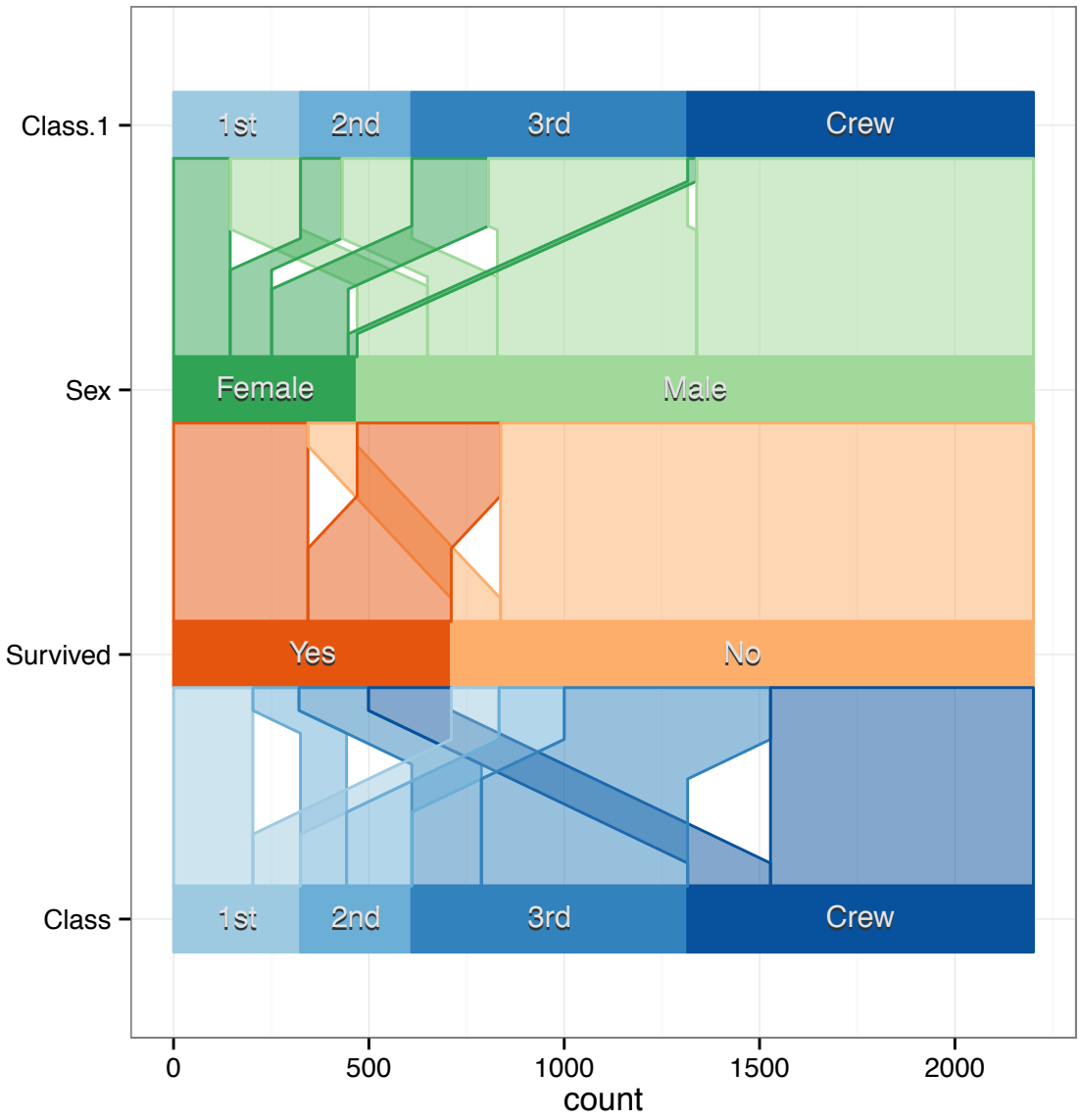


Figure 2.8: Common angle plot of the Titanic data.

2.5 Usability Testing

2.5.1 Test

To determine the effectiveness of the common angle plot, we conducted a user study in form of a survey asking participants to provide responses regarding the structure in two data sets with predominantly categorical variables. The Titanic data includes class, sex, age, and survival status for each person on board of the Titanic **?**). The gene data was retrieved from the UCSC Genome Browser Kent et al. (2002) and includes chromosome location for genes involved in one of three metabolism pathways: steroid biosynthesis, caffeine metabolism and drug metabolism. For each data set, participants were asked to provide responses for three tasks that analysts routinely perform as part of exploratory data analysis:

Task I: simple comparison task, chosen to be unaffected by any illusion. Performance on this task should be comparable across designs.

Task II: simple ordering, involving three pairwise comparisons, some of which are affected by the line width illusion or its reverse.

Task III: more complex ordering task with at least six pairwise comparisons, some of which are affected by either illusion.

Each participant was presented with two of the three types of displays. For each display, the participant was asked to complete each of the three tasks for each data set. All participants were evaluated using the same set of questions with multiple choice options as detailed in Appendix 2.7, regardless of display type or order. Participants were all shown the Titanic data first, then the gene data. This resulted in a crossover design as shown in Table 2.2 allowing for comparisons of display types and tasks while simultaneously adjusting for individuals' different skill sets and learning effect.

2.5.2 Results

We are investigating three different aspects of the experiment in this section: first, assessing general performance on the tasks according to percentage of correct responses, second, inves-

Titanic Data	PS	CA	H	PS	CA	H
Genes Data	CA	PS	PS	H	H	CA
#responses	8 (9)	6 (7)	8 (9)	6 (7)	10 (11)	8 (8)

Table 2.2: Overview of study design and participation numbers. The number in parenthesis indicates the number of participants completing the first block, but not the second.

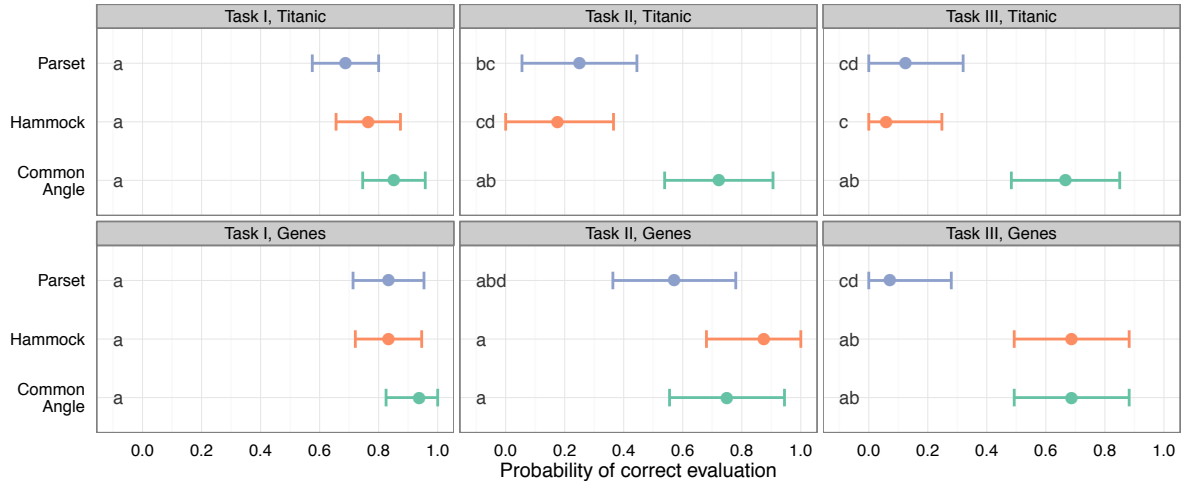


Figure 2.9: Overview of performance across tasks and designs. Points show average performance of subjects on each of the tasks, lines represent 95% confidence intervals adjusted for multiple comparisons. The letter at the front of each panel allow for an evaluation of significance of pairwise comparisons: if two averages do not share a letter, they are significantly different at a level of 0.05.

tigate the extent of variability due to subject-specific abilities, and finally exploration of the space of answers for the more complex ordering Task III.

Correctness of Answers

Answers for each survey question were recoded in binary form according to correctness (with 1 for correct answers, and 0 otherwise). This forms the basis for the evaluation of performance of the different designs.

Table 2.3 shows percentages of correct answers for each question under each design. Bold numbers indicate significantly different (worse) performance of a design compared to the common angle plot based on a generalized linear model with random effects to adjust for individuals' abilities. The model explains 69.5% of the total variability, corresponding to a highly significant deviance of 435.8 (p-value $\ll 0.0001$).

The observed results are in line with our expectations: as we aimed for, task I does not show any significant differences between the designs and has overall the highest percentage of correctness reflecting its low difficulty level. Generally, difficult levels seem to increase with complexity of the tasks.

parallel sets were affected the most by the line width illusion and show significantly worse performance for tasks II and III in the Titanic data, and for task III in the genes data. The performance on task II in the genes data is borderline non-significant, but shows a negative trend. Hammock plots led to significantly worse performance than common angle plots in the two questions that were affected by the reverse line width illusion, while they show equal performance as common angle plots for the other questions. For task III in the genes data, hammock plots have the overall best performance across designs– but this does not constitute a significant improvement over the performance of the common angle plot. Figure 2.9 gives an overview of performance of each design on all tasks.

Task	Data	Design		
		CA	H	PS
I	Titanic	85.2 (0.66)	76.5 (0.84)	68.8 (0.98)
	Genes	93.8 (0.51)	83.3 (0.78)	83.3 (0.90)
II	Titanic	72.2 (2.56)	**17.6** (2.31)	**25.0** (2.80)
	Genes	75.0 (2.80)	87.5 (2.13)	**57.1** (3.67)
III	Titanic	66.7 (2.69)	**5.9** (1.43)	**12.5** (2.13)
	Genes	68.8 (2.99)	68.8 (2.99)	**7.1** (1.91)

Table 2.3: Percentages (standard deviation) of correct responses for each task and design. Bold numbers indicate significant difference from common angle plot performance.

Individuals' skill levels

Figure 2.10 shows an overview of the predicted skill for each participant under the model. Skills are quite varied between -1.52 and 1.34, but a Kolmogorov-Smirnov test does not show significant deviation from a normal assumption (p-value 0.089). On the scale of the dependent variable the range in individuals' skills translates to a $17.5 = e^{1.34-(-1.52)}$ fold increase in the probability of answering a question on the survey correctly between participants with the best

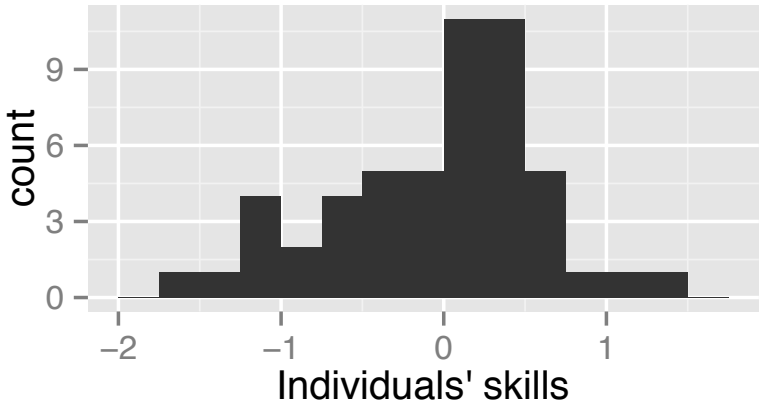


Figure 2.10: Histogram of the predictions of subject-specific skills.

skill set and the worst.

Evidence for line width illusions

Task III for the Titanic data required participants to order class levels according to the number of survivors, fewest to highest.

There are $4! = 24$ distinct orderings of the levels, corresponding to all permutations of length four. Some orderings are closer to one another than other orderings. The Cayley distance allows us to quantify this distance: the Cayley distance between two orderings is defined as the smallest number of switches necessary to get from one ordering to the other. Visually, this corresponds to a graph: each node represents one ordering, and two nodes are connected by an edge, if only a single switch is necessary to move from one ordering to the other, i.e. if the Cayley distance between these nodes is one. This results in a regular graph of degree six, i.e. each node is connected to six other nodes. Between any two nodes, the Cayley distance on the graph is equivalent to the length of the shortest connecting path between the two nodes. Figure 2.11 shows an overview of the permutation space together with an overview of the survey results.

The colored dots on top of the graph correspond to the responses from the survey. The size of these dots is proportional to the number of observers choosing this particular ordering. It becomes obvious from the three graphs in figure 2.11 that the answers to different designs

occupy quite different regions, while answers based on the same design are quite close – usually separated by only one edge.

The correct ordering, as well as the orderings assuming the line width illusion and its reverse are marked by symbols. Answers for the common angle plot are centered around the correct answer, while responses to parallel sets cluster around the response corresponding to the line width illusion. Due to the smaller number of responses to the hammock design a clear clustering of the answers is not recognizable, but the answer for the reverse line width illusion is among the responses. Table 2.4 gives an overview of all responses to task III for the Titanic data.

Order	CA	H	PS	
Crew, 1st, 3rd, 2nd		2		
2nd, 1st, 3rd, Crew		6		reverse line width illusion
Crew, 3rd, 1st, 2nd		1	1	
2nd, 3rd, 1st, Crew	12	7	1	correct
2nd, 3rd, Crew, 1st	1	1	2	
Crew, 3rd, 2nd, 1st	2			
3rd, 2nd, Crew, 1st	1			
1st, 2nd, 3rd, Crew	1		1	
1st, 3rd, Crew, 2nd	1		2	
Crew, 2nd, 3rd, 1st			6	line width illusion
2nd, Crew, 3rd, 1st			3	
Total	18	17	16	

Table 2.4: Responses to task III in the Titanic data: order levels of Class by the number of survivors (smallest to largest).

Common angle plots show the best performance in terms of correctness (66.7% on 18 responses), compared to a correctness of 12.5% for parallel sets plots on 16 responses, constituting a significantly better performance of common angle plots at a level of 0.0027, based on a Mantel-Haenszel test (the difference in performance to hammock plots is also significant at a level of 0.0006; but there is no significant difference in correctness between hammock plots and parallel sets.). While the intuitive assessment of lines by their width orthogonal to their direction is well known, it is surprising to see its strength: in this particular setting, it is strong enough to 'shrink' the horizontally widest line for six out of 16 participants by at least 44%, from 212 to below 118, and a further three participants perceived a shrinkage to below 178, a distortion

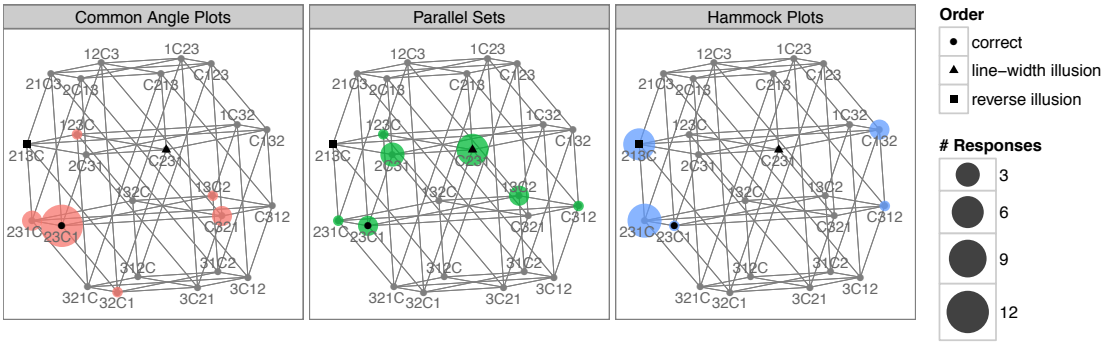


Figure 2.11: Answers to task III in the Titanic data – each node corresponds to a single ordering of the levels in variable 'Class'. Lines are drawn between orderings that are only one swap of levels apart. The colored dots show responses from the survey, their sizes depend on the number of responses for each ordering.

factor of at least 16%.

Opinion on common angle plots

Answers to the question of 'which chart did you like better?' are shown in table 2.5. There is a clear endorsement in favor of common angle plots versus the other two types of displays. The most common reason cited for the choice was a facilitated comparison of width, area or "size", The only consistent complaint against common angle was a preference for straight lines. This purely aesthetic preference is deeply rooted and in our opinion the biggest challenge for common angle plots.

			Which chart did you like better?	
			Chart 1	Chart 2
PS	vs	**CA**	2	**6**
CA	vs	PS	**4**	2
H	vs	**CA**	3	**5**
CA	vs	H	**8**	2
H	vs	PS	3	5
PS	vs	H	1	5

Table 2.5: Preferences for first or second chart across all six combinations of questions and chart types.

2.5.3 Methods

The survey was created using the Qualtrics Labs, Inc software (www.qualtrics.com). For survey contents, see Appendix 2.7. The study design is presented in section 2.5.1. All models are fit in the `lme4` package Bates et al. (2012) within the software frame work of `R` 2.15.1 R Core Team (2012). Comparisons are adjusted for multiple testing using the `multcomp` package Hothorn et al. (2008) and evaluated for pairwise significances using the `effects` package (Fox, 2003), (Fox and Hong, 2009).

2.6 Discussion

For data with a large number of variable levels, common angle plots may introduce more line crossings than hammock plots, while the number of crossings is the same between parallel sets and common angles. This may affect the effectiveness of the overall display. Use of color may also be problematic with many variable levels, as readers may have difficulty resolving a palette of colors separated by small intervals. Certainly, resolving many colors is difficult for audience with a history of color blindness and may be technically limited in print applications. The issue of color is consistent regardless of display choice between parallel sets, hammock plots or common angles. Further study is necessary to resolve the potential for distortion in different application, especially the impact of bands displayed with extreme values for θ, a known source of perceptual inaccuracy Heer and Bostock (2010).

One opportunity for improvement lies in the algorithm to determine thickness of the connecting line. In the tested version of common angle plots, the line width was not explicitly defined - the line width is a byproduct of specified θ for a band that connects marginal bars. Using equation 2.1 to determine the band thickness while keeping θ consistent across all bands is a common angle-hammock plot hybrid that bears further investigation. A drawback of this approach is that sum of ribbon w_h will no longer match widths of marginal bars. This may create a additional processing burden placed on the audience to map the relationship between band width and the with of marginal bars. Both hammock plots and this modification of common angles face the issue of band *area* as context to support reader interpretation of w_o. Since

the band area is now related to the incident angle θ, changes in the display aspect ratio may have a distortion effect. In the study described in this paper, this effect was not evaluated and aspect ratio was kept constant.

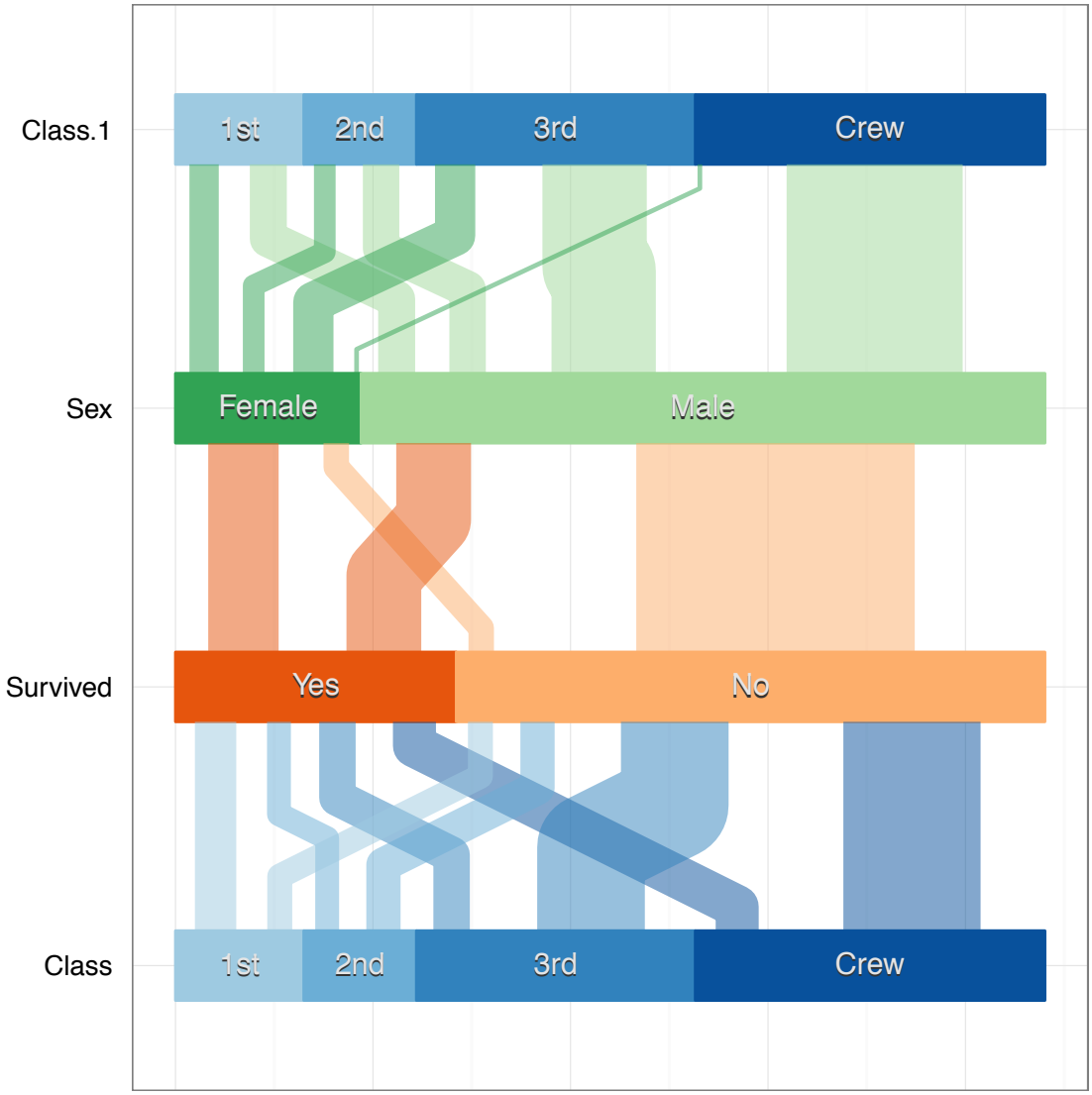


Figure 2.12: Common angle plot of Titanic data using hammock correction.

In the original paper, parallel sets were introduced to reflect a hierarchy of variables. Prior examples in this paper show sets of two-dimensional plots to focus on the association between pairs of variables. With color coding, it is possible to show hierarchies in all of the types of displays. Figure 2.13 shows a common-angle plot with a hierarchy: survivors of the disaster

are marked in blue, non-survivors by orange. From top to bottom of the plot a hierarchy is drawn, considering first survival, then gender, followed by age and finally class membership. The coloring tracks survival status throughout the hierarchy, the layout in a common angle plot makes comparisons valid across all levels. This is of particular importance in hierarchical displays, which by definition have a larger number of smaller groups than displays without a hierarchy exacerbating problems induced by the line width illusion.

Another opportunity for extending common angle plots is to add interactivity. It is important to note that any additions of functionality via interactivity should not come at the expense of developing distortion-free displays. Simply augmenting a plot that has distortion of the *line width illusion* variety does not eliminate the presence of that illusion, rather it obfuscates the message of the display. A display with visual cues in conflict with the interactive feedback introduces cognitive load by asking the audience to decide which source is correct. In the case where interactive feedback (e.g. summary data on mouse hover) is accurate when the visual cues suggest alternate interpretation, a mistrust of the graphical presentation may develop. In an extreme case, the user may develop a mistrust of their own perception, which would reduce effectiveness of all data displays, regardless of the presence of any perceptual distortion. In the alternate case, where audience chooses to rely on perception over interactive feedback, it is possible that distortions in displays that are part of initial exploratory analysis may lead to research choices that are unsupported by data. This is both a waste of resources and, when human or animal subjects are involved, may lead to ethical violations.

2.7 Conclusion

We have proposed a new chart type for visualizing multivariate categorical data, common angle plots, and tested its usability compared to existing charts that perform a similar function. Results from user testing indicate that common angle plots effectively communicate underlying data without encouraging perceptual distortion of the *line width* illusion. Two other chart types which address visualization of multivariate categorical: parallel sets and hammock plots, are subject to the line width illusions due to contextual framework. Audiences perceive parallel sets with distortion due to a natural tendency to evaluate line width in the orthogonal direction

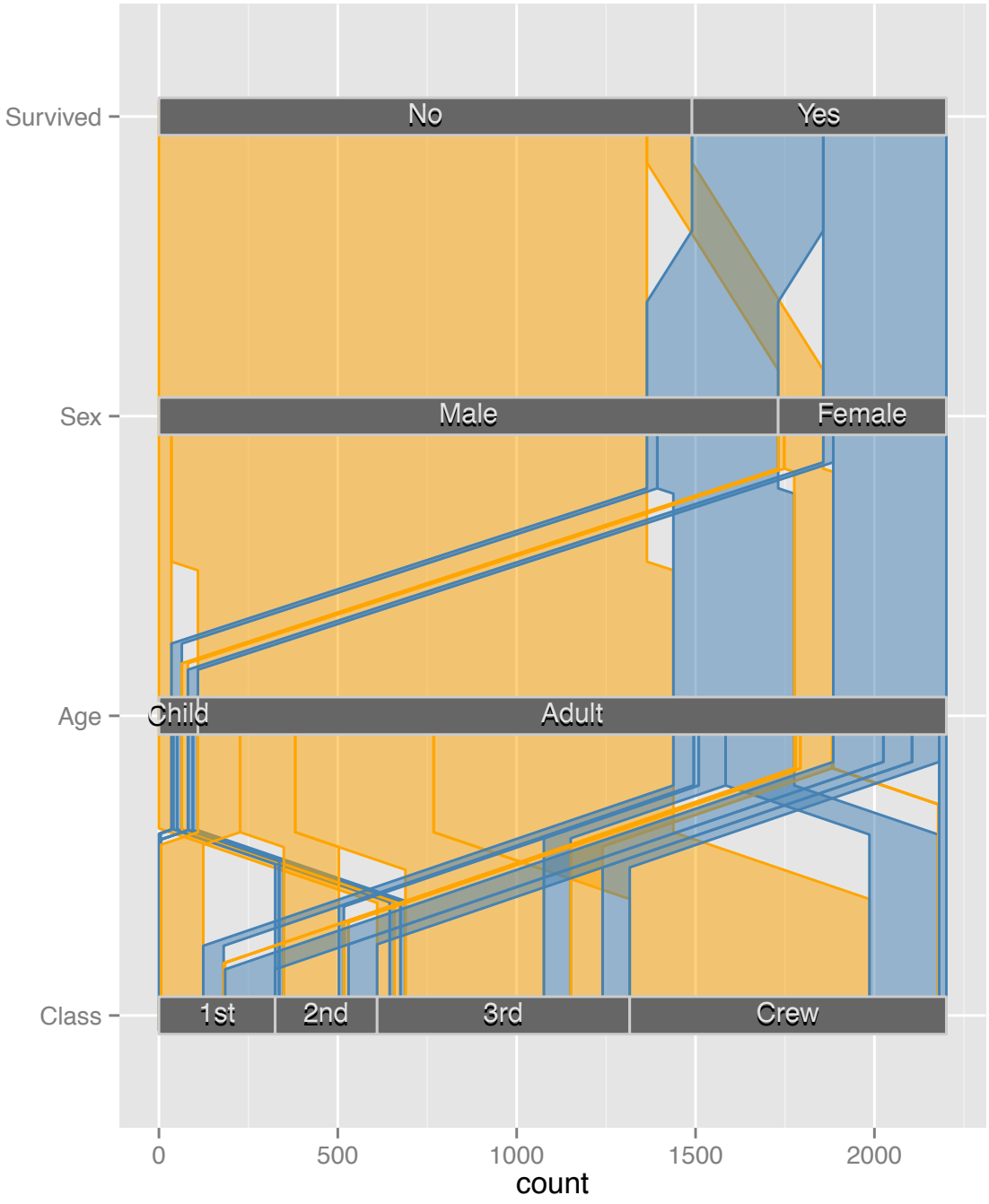


Figure 2.13: Common angle plot of the Titanic data using a hierarchical structure in the variable (cf. to parallel sets chart in Davies (2012)).

while data is mapped to the horizontal width. For hammock plots a correction is made to map data to the orthogonal width, however the centered line intersection with axes creates a strong contextual encourage evaluation of line width using the horizontal measure (*reverse line width illusion*). Common angles avoids the perceptual distortion associated with either version of the illusion regardless of the underlying data set.

Appendix

Survey

At the survey start, participants were presented a brief tutorial regarding the different plot types. The tutorial can be found at http://mariev.net/tutorial.html

The survey consists of two blocks of questions each pertaining to one data set (Titanic data or gene data). Each block was presented with a single plot to use as reference when responding. Two different plot types were shown for the two different blocks, yielding a total of six unique orderings of plot types as shown in table 2.2.

Participants were randomly assigned to one of these six combinations. This study design structure was imposed, in part, to encourage participation by reducing the amount of time for survey completion. Completion of all survey questions was anticipated to take 10 - 15 minutes. No personally identifiable information was collected, nor was any compensation offered.
The questions pertaining to the Titanic data were:

Task 1: *Agree, Disagree or Don't Know/Can't Determine with the following statements:*

- There were an approximately equal number of Male and Female Survivors
- The group with largest number of travelers was Female Survivors
- There were more Male Non-Survivors than number of males in First and Second Class Combined

Task 2: *Order the following groups by number, fewest to most*

- 1st Class female passengers

- Male Survivors
- Crew Survivors

Task 3: *Order the categories of Class by number Survived, fewest to most.*

- 1st
- 2nd
- 3rd
- Crew

The questions pertaining to the gene data were:

Task I: *Agree, Disagree or Don't Know/Can't Determine with the following statements:*

- There are about the same number of genes in the group "steroid biosynthesis:chromosome 1" as in the group "caffeine metabolism: chromosome 8"
- The group with the greatest number of genes is "drug metabolism:chromosome 4"
- there are more genes involved in the group "drug metabolism: chromosome 1" than all genes involved in the caffeine metabolism pathway

Task 2: *Order the following chromosomes by number of genes involved, fewest to most.*

- steroid biosynthesis :: chromosome X
- steroid biosynthesis :: chromosome 4
- drug metabolism :: chromosome X

Task 3: *Order the following chromosomes by number of genes involved in steroid biosynthesis pathway, fewest to most.*

- chromosome 1
- chromosome 4
- chromosome 8
- chromosome X

Participants' demographics

All students, staff and faculty from Iowa State University programs in Statistics, Bioinformatics and Computational Biology and Human Computer Interaction were invited to participate by email. 93 individuals accessed the survey; 86 participants gave consent, 15 of those dropped out right after, 20 went to the training site and did not return. Out of the remaining 51 participants, 46 individuals submitted responses for all questions and five gave responses to the first block of questions.

Participants used their own personal computing devices to access the survey, a majority of participants used Intel Mac OS X (versions ranging from 10.6.8 to 10.8.2), while Windows was the next most common operating system. The preferred choice of browser was Firefox, followed by Chrome. For two participants, the Qualtrics survey software was unable to capture operating system or browser information.

CHAPTER 3. DEVELOPING WORKFLOWS FOR BUSINESS AND STATISTICAL AUDIENCES: A CASE STUDY AT USDA APHIS

A paper in submission to R Journal

All technical tools, code, writing and analysis is my original work.

Abstract

Statisticians working in R often face the unruly situation of input data provided by collaborators being improperly formatted, a downstream audience that is completely uninterested in integrated reporting tools, and/or both. Multiple packages exist in R to address the issue of data cleaning, and workarounds to track changes in LaTeX documents are plentiful. Statisticians directly employing these methods are not focusing on the underlying issue: how to transform collaborators into useRs - even when organizational culture or resources prevent directly working in R. As a solution, we propose using technical tools developed using human-centered design principles which consider both the statistician and collaborators. This process directs collaborators in the performance of data cleaning and report generation tasks that are consistent with techniques used within R. The statistician is freed to perform analysis. In a case study of USDA APHIS CVB, we see the impact on turnaround times and organizational culture.

3.1 Introduction

R users often face an unruly situation when collaborators are unfamiliar with the structure and syntax of fundamental data objects and methods in R. Although it may seem intuitive for analysts to format information as a `data.frame` with each variable encoded as a column and

each record as a row and with missing values designated `NA`, this structure does not support best practices for error-free data entry or an eye-readable display. If a collaborator's mental paradigm of data structures is consistent with the structure displayed on graphical interfaces supporting daily activities of data entry and reporting, serveral challeges arise. First is simply that the R user must clean and reshape the incoming data, a task that is recognized as necessary but tedious (Wickham, 2007, 2010b). Second is the obstacle of communicating between analyst and collaborators when investigating discrepancies - each party must translate repeatedly between multiple representations of the same information. Finally, there is the concern of efficiency: the translation of data between collaborator and analyst is not a value-added component of the overall analysis, it is merely an activity to facilitate the process. Likewise, the default technique of embedding R code into PDF files via `Sweave` and `knit` that many R users rely on for reporting is not correlated to the interface that business audiences are accustomed to, which allows for management of collaborative editing of report text (e.g. tracking, accepting and rejecting proposed changes) from within the document.

A tempting solution is to require that all stakeholders involved use R. Unfortunately, working groups that bring together collaborators of great diversity may not have the resources (e.g. time, local access, inclination, training) to aquire proficiency in R. So, is answer then to charge the statistician with all tasks - data preprocessing, analysis, report generation? **No**, this is a poor practice which encourages a conceptual paradigm that is inconsistent with the division of work, reducing accountability. That is, "good" data submitted by upstream collaborators is not "good enough" for use in R (Figure 3.1a), but the work of data cleaning and restructuring is not accounted for. A preferable solution is to develop tools from the perspective of user-centered design, allowing statisticians, scientists and non-scientist decision makers to perform tasks in a manner consistent with techniques in R, but from a working environment that considers individual expertise. Collaborators may then take on greater responsibility for data cleaning and report generation tasks, leaving the statistician to focus on analysis.

The user-centered approach addresses the problem at its core: *how to communicate the needs of statistical analysis to business stakeholders*. By developing tools that make use of interafaces most familiar to a user subgroup, we remove the obstacle of technical skill acqui-

sition. These tools facilitate extension of the R user group to include individuals not directly programming in R. In this paper we describe the process behind creating such tools in R to facilitate task performance associated with data analysis across working groups. We see, via a case study at USDA APHIS CVB, how this process reduces turnaround times and positively affects organizational culture.

3.2 Preparation: defining the workflow

We borrow from the field of business productivity to define the workflow as a first step, which is consistent with implementing Lean in a production environment (Womack et al., 2007). Defined workflows are frequently used to formalize and coordinate multistep processes shared across multiple individuals and/or working groups. While we implicitly use workflows on a daily basis, moving from an unspoken workflow towards one explicitly defined presents opportunity to retain knowledge at an organizational level and allows for capturing quantitative metrics to ascertain quality.

Workflows that apply to data transformation and incorporate computational or bioinformatics methods are a special implementation, a *scientific workflow* (Figure 3.1c). The level of detail that is sufficient for business process modeling merely is an abstraction that highlights main task(s) by stakeholder (Hettne et al., 2012) (Figure 3.1b). For each stage of the scientific workflow, three parameters exist:

1. Input description
2. Transformation rules
3. Output description

Output of one stage becomes input for the next stage. This structure is consistent with the definition of a *function* (**?**), and thus each stage is a candidate for replacement by a technical solution. For stages involving statistical analysis a clear candidate for implementation lies in R. Subsequent Lean improvement steps target reducing turnaround time, eliminating redundancies, reducing the number of errors, all of which are relevant paradigms for iterative revisions of

functions implemented. In addition to the process performed at each workflow stage, defining the workflow includes identification of the user group that is responsible for each stage which allows the developer to identify what working environment and design considerations will have positive impact.

3.3 Upstream collaborators: handling data cleaning

In the transition from inaccurate to scientific workflow (Figure 3.1) we can see that upstream collaborators are now tasked with the stages affecting data entry, cleaning and reshaping to generate a data structure suitable for statistical analysis. In many cases, these stakeholders do not have the same perspective of "raw data" as does an analyst. What is seen as an initial step that takes place prior to analysis may be in actuality be the final output step of a different workflow. For example, biologists may view tables as a high level summary of testing observations - certainly not the raw data that is preserved in the ubiquitous lab notebook. This may require stronger cues within the tool design to communicate acceptable data types (i.e. specifying "numeric" is insufficient, there must be instructions regarding how to report censored data).

Initial recording of observed data may also to environmental limitations. Recording observations by hand on a worksheet may be a reasonable method for some types of fieldwork, which requires error-prone manual transfer for electronic capture. Even when electronic entry is employed in the field, the interface is designed to reduce entry issues but with a very specific export format that may or may not be coordinated with the needs of analysis within R (Harris et al., 2009; **?**). A common example is use of white space to represent both the case of multiple entries of the same value in a graphical display of the table *and the case of omitted information (e.g. "NA").* When using default import functions in R, no distinction is made between empty cells that hold no information (true NAs) and empty cells that act as pointers to the cell directly above (false NAs).

Another obstacle to consider are differences between user groups performing entry versus that of individuals performing analysis. We have previously mentioned that entry may be performed in the field, but even from traditional office workspaces, local IT policies and protocol

may prevent installation of R. In this case, tools developed in R - no matter how elegant - will not be used. Even when use policies allow installation of R on local systems, the Technology Acceptance Model (TAM) (Davis, 1989; Venkatesh and Davis, 2000; Venkatesh, 2000; Venatesh et al., 2003; Venkatesh and Bala, 2008; Bagozzi, 2007) suggests that there are many factors motivating individuals to adopt use of R. Keeping this in mind, for many users the interface with greatest perceived ease of use *is not R.*

While these obstacles provides motivation for developing entry interfaces that may be used independently from R, there is still opportunity to incorporate data paradigms from R into the chosen interface. To avoid the situation of ambiguous meaning associated with empty cells, timely prompts and error checking can provide immediate corrective feedback. Examples and documentation automates guiding users through the process of best practices in data entry. One might question the incentive for development for environments outside of R: simply, an inclusive attitude allows highly skilled R users to offload routine tasks, since a well-developed interface allows for self-direction. Another opportunity lies in reducing the perceive ease of use for future implementations of R. That is, using the external interface exposes current R non-users to underlying data handling concepts, which makes a later transition to R incrementally less challenging.

There are certainly instances where automated approaches have minimal, if any, impact. For example, if submitters encode variables over multiple data columns by using a so-called informative headers, e.g. columns titled “day 1 body temperature”, “day 2 body temperature”, etc instead of limiting the entry to two columns (e.g. “day” and “body temperature”). Although only two pieces of information are described, the number of columns describing that data varies based on the duration of the study. This poor practice leads to data tables that grow in both dimensions over the course of the experiment, and determining simple summaries becomes computationally inefficient.

When expert review is needed to evaluate data quality, it is possible to reduce the effort required by creating custom import and summary functions in R that display information in an eye-readable manner designed to highlight common error sources. The combination of fully automated methods with manual inspection allows stakeholders to determine what tasks are

the greatest value added - programming robust automated methods for highly repetitive tasks or assigning individuals to unique activities that may not warrant up-front resource costs.

During data input, it is not enough to have tools that are technically *capable* of capturing complete information. If statisticians wish to hand off routine data cleaning and shaping tasks to other stakeholders, the choice of technical tools must be accessible and enable self direction when creating data objects that transition easily into R. Doing so allows for all users to focus efforts in a manner that is value-added while omitting rework.

3.4 Downstream collaborators: reporting

Another stage which requires consideration of diverse user groups is when reporting findings to audiences. Although it is trivial to create reports from R to pdf files using `knit` or `Sweave` in conjunction with LaTeX these output files may not be best suited for collaborative reporting. Of particular challenge when collaborating using LaTeX documents is identify contributions by different authors for review and acceptance/rejection. Using additional LaTeX packages such as TrackChanges (Salfner and Pablant, 2009) or custom commands requires that all collaborators have proficiency with and access to LaTeX . For some user groups this may be an unsurmountable obstacle - the default state is for the statistician to take on duties of tracking and compiling changes, a significant effort that takes away from core tasks.

Pragmatically, it makes sense to move away from R and LaTeX after analysis is complete but before any collaboration occurs. That is, creating reports within the confines of R using built-in access to LaTeX normally makes sense for expert R users because it eliminates a handoff between R and alternative reporting software. In this scenario, it makes sense to undertake the handoff sooner rather than later, in order to make collaboration technically accessible to a wider audience. The cost of this handoff in resources (e.g. time, custom scripts) is more than made up (from the statistician's perspective) because it shifts the role of managing collaborative edits away from statisticians. Pandoc provides a ready method of automatically translating between multiple file formats, with the greatest flexibility for files initially in the markdown syntax. As in the input scenario, it is important to consider the existing skill set of the user group when selecting the final format.

3.5 Analysis: Other opportunities

Development of R packages presents opportunities for optimizing the analysis portion of a workflow. With input data highly structured, it may make sense to wrap frequently-used steps into a single function. Custom analytical methods are also a rich target for package development. Regardless of the tasks achieved using custom packages, the value of development is that it further formalizes the analytical process and requires consensus between individuals performing analysis.

This is valuable when working in regulatory environments, where procedures are subject to validation, auditing, and documentation requirements. Package test cases can act as validation. Code and help pages provide documentation. Auditing becomes an activity of loading the correct package (including version) and running the R code chunks which are defined during reporting. Additionally, concensus and consistency are core missions in this field. By linking package methods to tasks specified in the workflow, it is possible to ensure that data of the specified input gets processed in the same manner. Graphical displays and typesetting may be standardized (e.g. font, fontsize, color palette, figure width, resolution, margins, etc.). It is also clear from the R script when deviations from a standard processing occur.

Even in fields that do not face regulatory oversight, consensus on a standard for analysis presents downstream audiences with a consistent message. Even simple choices, such as how many digits to round data has informative value. Of greater impact are the selection of summaries to display and parameters (or range of parameters) which are implemented. The formalization process, and documentation that custom functions provide, mean that downstream audiences spend less time seeking to elucidate details of the analysis process, and are able to immediately focus on results. Well-designed visualization methods also offer the benefit of acting as a benchmark to provide examples inspiring future work.

3.6 Case study: USDA APHIS CVB Statistics section

One mission of United States Department of Agriculture (USDA) Animal and Plant Health Inspection Service (APHIS) Center for Veterinary Biologics (CVB) Statistics Section is evaluat-

ing veterinary biologics for licensure in a manner compliant with provisions of the Virus-Serum-Toxin Act (VSTA). Title 9 of Code of Federal Regulations (9CFR) specifically stipulates that products falling under licensing requirements receive approval only after a successful evaluation of records and methods used to demonstrate validity of claims. No mention exists of submission format at a data management level. Practically, this policy means statisticians in CVB must accept submissions in a wide variety of formats including output from multiple software sources the section does not have access to, and occasionally, hard-copy reports.

3.6.1 Revising the workflow

The workflow before implementing the process described by this paper was labor-intensive and confrontational. Firms were transforming internal data structures specifically for submission to statisticians at USDA. After recieving data, statisticians often had to extensively reformat the data simply to enable importing into R, only to discover that submitted information was incomplete for analytical purposes. When a statistician contacted the firm requesting clarification, it was often a struggle for firms to understand what was missing. This confusion was exacerbated because the format a statistician was referring to (usually multiple tables of "long" or "wide" layout) was different from a firm's original submission. Updated information from the firm was resubmitted using whatever original format initially provided. The statistician would have to step through the entire reformatting proceedure to determine if an updated submission was complete. Multiple iterations of this process was necessary in complex cases and management of file versions became a non-trivial concern.

3.6.2 Input

3.6.2.1 Data Cleaning

A computational solution to this problem is one which automates as much of reshaping and data checking as possible. Ideally statisticians should only get involved when evaluating whether a data set is complete for analytical purposes. At first we intended to implement all computational tools in R, but installation and operating R even at extremely trivial levels was

a hardship for many firms. What has emerged after multiple rounds of user testing is a set of Excel templates that walk users at firms through the process of entering data. Templates are freely distributed on a public website maintained by CVB at http://www.aphis.usda.gov/animal_health/vet_biologics/vb_data_formats.shtml. Each template type is distributed as a zip file containing an empty template, a template containing example data, examples of output files generated (.csv) and a instructions regarding how to use templates in a Word document. Templates are Excel workbooks where each required data table is set up on separate worksheets. A text box on each sheet guides users through the process of filling out tables, and includes definitions of what are acceptable values for each variable. The instruction text box updates in real time to prompt users based on current state of data entry. Drop-down lists of acceptable values for variables are included when possible, and data entry is structured to minimize duplicate entries (e.g. IDs for a 96-well plate needs only be entered once and this value is automatically applied to records corresponding to all 96 wells). The final step of data entry using the template is to press a macro button for exporting into csv files of "long" format preferred by statisticians. This macro button also checks for common errors, such as cells containing wrong data type, empty cells or duplicate entry and alerts users of corrective action needed. Output files for submission cannot be generated without passing error checks. One worksheet of the template is a resource for troubleshooting including links to online resources, contact information for CVB and version information. *The purpose of these templates are to act as a familiar graphical interface guiding non-statisticians independently performing data entry of information for a statistical audience.* With implementation of the template framework, CVB has gone from returning 50% of data submissions as incomplete - requiring follow-up by firms - to $> 98\%$ compliance with formatting expectations.

3.6.2.2 Importing and Checking for Data Completeness

The next stage of the updated workflow is to import data into an R session for downstream use by statisticians for analysis and report generation. The *input description* is the structure of files generated by templates, the previous stage. The *output description* is via the definition of reference classes in **dataFormats** (Vendettuoli, 2012), unique to each template type. *Trans-*

formation rules are wrapped in import functions of **dataFormats** and rely on `read.csv` and `new` along with `$initialize` and `$validate` methods of the reference class. Documentation for all three parameters of this workflow stage is accessible at a detailed level in code and at an abstracted level via R help pages which include examples.

Including validation criteria in reference class definition achieves several purposes. First is to address the issue of clean management of hand-offs between workflow stages. That is, it is possible for unintended changes to template outputs to occur if files are incorporated into other workflows. For example, a submitting firm may choose to visually inspect contents of template export files and in the process indadvertently make changes to data *after* template error checking proceedures are completed. Alternatively a submitting firm may use surrogate software to generate csv files. Validation that imported files pass same criteria that template exports are subject to ensures that the data structure that firms believe they are submitting is consistent with what statisticians have access to in their active workspace. Validation offers opportunity to provide error messages to aid troubleshooting. For example, attempting to import a partial set of data files generates the error message 'noligssampximportFromELISATables: Need all four ELISA Format tables.

Additionally defining complex data as a single object creates an opportunity to deploy generic functions which reduces documentation that an individual statistician needs to record while performing analysis, particularly when performing repetitive tasks. Specifically in the stage of checking for data completeness, human intervention is necessary to verify if submitted data includes all necessary variables. To perform this task, comparison needs to be made between written description of experimental design and submitted data. For submissions of format 'noligssampxgeneral, these values are stored in field 'noligssampxvar.

3.6.3 Analysis: Initial Summaries

Defining generic functions also facilitates exploratory analysis processes. **dataFormats** includes the generic functions `getTable` and `getPlot` which use functions in **xtable** (Dahl, 2013), **grid**, and **ggplot2** (Wickham, 2009) to produce graphs and plots customized for each data format. As with the importing workflow stage, documentation of input, output and transfor-

mation rules is captured through code and help pages, with examples. Beyond the immediate application of quickly creating tables and charts from well defined inputs, these generic functions present an opportunity for statisticians to explore the capabilities of package dependencies without the resource cost of learning the syntax of underlying methods. While this approach has limited flexibility, it can act as an incentive to encourage acquisition of fundamental skills for user groups who do not prioritize pursuit of programming competencies, by providing practical examples and enabling high-level use at outset.

It is important to note that structure for these default plots and tables were determined via collaboration between statisticians. Beyond simply improving documentation, display consistency and efficiency the use of these summary functions represent a consensus regarding initial analysis. Presenting a consistent message allows downstream audiences to focus on the data at hand instead of interpreting idiosyncratic differences between different statisticans. It also allows for smooth transistions in cases where submissions are passed from one statistican to another. Consistency ensures that the same criteria is applied to all firms

Not only are these tables and plots available individually via **dataFormats**, they are employed by the `createReport` function in **CVBreports** (Vendettuoli, 2013) to generate automated initial pdf reports using knitr and pandoc.

```
require(dataFormats)
require(CVBreports)
x <- importFromELISATables(paste(system.file('data', package = 'dataFormats'),
  'ELISAExample', sep = '/'))
createReport(x, template = 'basic_elisa_html', page_title = 'mypage',
  load_data = 'elisa.rda', ext_files = 'figure')
```

The above code, along with `sessionInfo()` is all the documentation required for a submission that conforms to template formatting for a preprocessing report of type 'noligssampxpotency (Figure 3.3(b)). The output of this a LaTeX file which includes a table displaying plate level meta data color coded tables showing the dilution and optical density (OD) values color coded by layout group for each plate. Tables allow reader to quickly see the position of a

particular layout group or dilution scheme on the plate. Color coding matches plots displaying the OD of each layout group compared to the independent dilution variable.

3.6.4 Output: Reporting

While the above example is useful in very defined circumstances, it is reasonable that statisticans may wish to generate reports that involve modeling and/or other analysis that may be unique to the submission. It is also practical to implement reporting solutions in a manner that allows easy extension to comply with changing needs. The engine underlying `createReport` is **CVBreports** which is a collection of templating and accessor tools.

3.6.4.1 Templates

Each template is a file of type ‘noligssampx.rhtml, ‘noligssampx.rmd, ‘noligssampx.rnw. Within a template, the placement of modular elements, such as the R code to import a data file, common headers and footers, etc are indicated in comments by name. When `createReport` is run, the placeholders are populated. Templates of type ‘noligssampx.rmd offer greatest flexibility, because of the option to transform output ‘noligssampx.md files into MSWord documents. Reviewers, who perform much of their daily duties using MSWord, are then able to modify the document directly and/or copy and paste text equations and figures without having to switch programs.

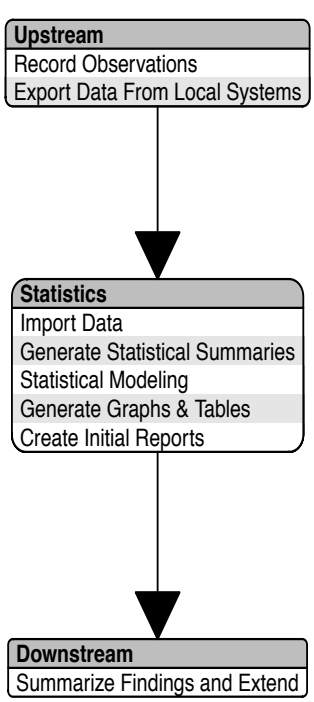
3.6.5 Impact: Quality & culture shift

Prior to the introduction of this revised workflow, the elapsed time between submission and initial data exploration was routinely on the order of 4 to 6 months, with the actual hands-on exploratory evaluation lasting two or more days. With the updated workflow, this initial exploration takes approximately 48 hours, with simple cases requiring less than an hour of hands-on analysis. Much of this improvement may be credited to prioritizing exploration of incoming data amid other duties. However, it is only with the transferance of data cleaning activities to submitting firms that the task of initial exploration now only requires resource costs that do not interfere with concurrent obligations.

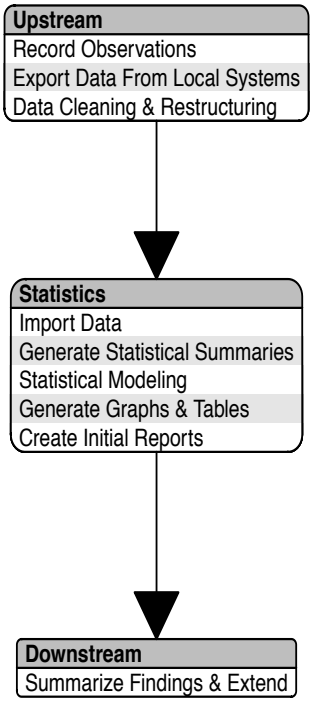
Potentially of greater impact is the coinciding culture shift. Prior to formalization of workflow at a scientific level, no one in the Statistics section considered the use of 'noligssampx.rnw files for reporting by use of `Sweave`. This was a lag of over a decade since the inception of `Sweave`. With **CVBreports**, the Statistics section has adopted tools in **knitr** less than two years after the package's initial beta release. Modularization has also led the section to group functions into packages by task. In addition to packages for workflow processing, **PF** (Siev, 2012) (for statistical analysis of prevented fraction), has been released to CRAN and other packages are in development. The focus on increased transparancy and reproduciblity also supports better coding practices. Most recently, the section is working to develop a local style guide to improve the readability of shared and archived scripts.

3.7 Conclusion

With the implementation of formatting templates, **dataFormats**, and **CVBreports** at CVB Statistics, we have described the process of extending R data handling and reporting paradigms to collaborators that typically do not have resources that allow interaction in R. This allows statisticians to focus on analysis while offloading responsibility for routine tasks of data input and reporting to collaborators. Additionally, creating such tools extends the influence of R paradigms beyond traditional R user groups. The steps of this process are: (1) Formally define the workflow (2a) Create interfaces for data entry that are consistent with strengths of upstream collaborators (2b) Define formal objects in R and import methods to handle data entry output (3) Develop R packages for custom analysis (4) Use `knit` and pandoc to generate output files that enable participation by downstream audiences. Implementation of this process empowers collaborators up- and down-stream of the statistician to be part of a workflow that implements R paradigms for data and report handling within existing user working environments. At the USDA APHIS CVB Statistics section, this process has significantly reduced turnaround times, increased organizational knowledge and allowed earlier adoption of new technologies.

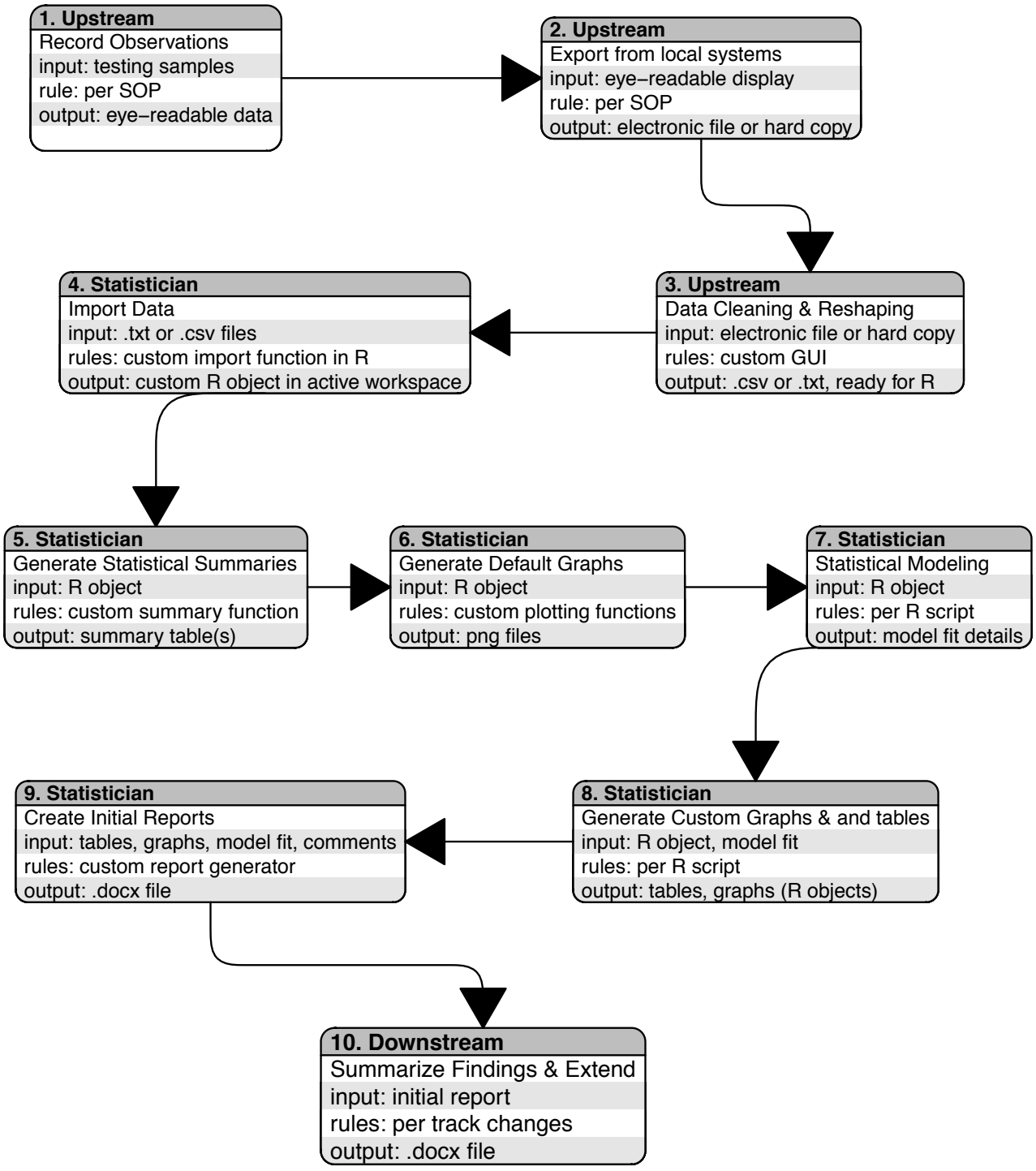


(a) Incorrect workflow model. The work of data cleaning and restructuring is left up to the statistician with the misguided belief that no data cleaning or restructuring is needed. Likewise, statisticians exert effort to manage collaborative input, which is necessary activity but not value-added. This workflow places responsibility for tasks that are conceptually accessible by collaborators with the statistician.



(b) Abstracted workflow. Stages of the workflow are broken down by stakeholder groups, but not all workflow stages are identified. The high level of detail allows for accurate assignment of accountability and resource allocation, but limited specification for tool development beyond specifying user group.

Figure 3.1: Generic workflows of differing quality or resolution. For each workflow stage, the user subgroup is designated in the header. Upstream and downstream user groups are non-statistical (business) audiences. For effective tool development, it is important to identify the workflow at the scientific levels. In many working environments defining the workflow may lead to parallel or iterative diagrams. Techniques presented in this paper are flexible to accomodate a variety of scenarios.



(c) Scientific workflow. Responsible stakeholder groups are identified for each stage, as well as the data input description, transformation rules and output description. This workflow acts as a roadmap for later tool development because it specifies user group - which determines working environment - as well as technical expectations. Best practices in modular programming encourages breaking down stages into the smallest unit possible and employing wrapper functions to engage multiple units in one group. In the scientific workflow breakdown, we see that stages 4 - 6 are an attractive target for wrapper functions if data is highly consistent (e.g. multiple submissions as part of the same longitudinal study). Custom" refers to tools selected or created for this particular workflow.

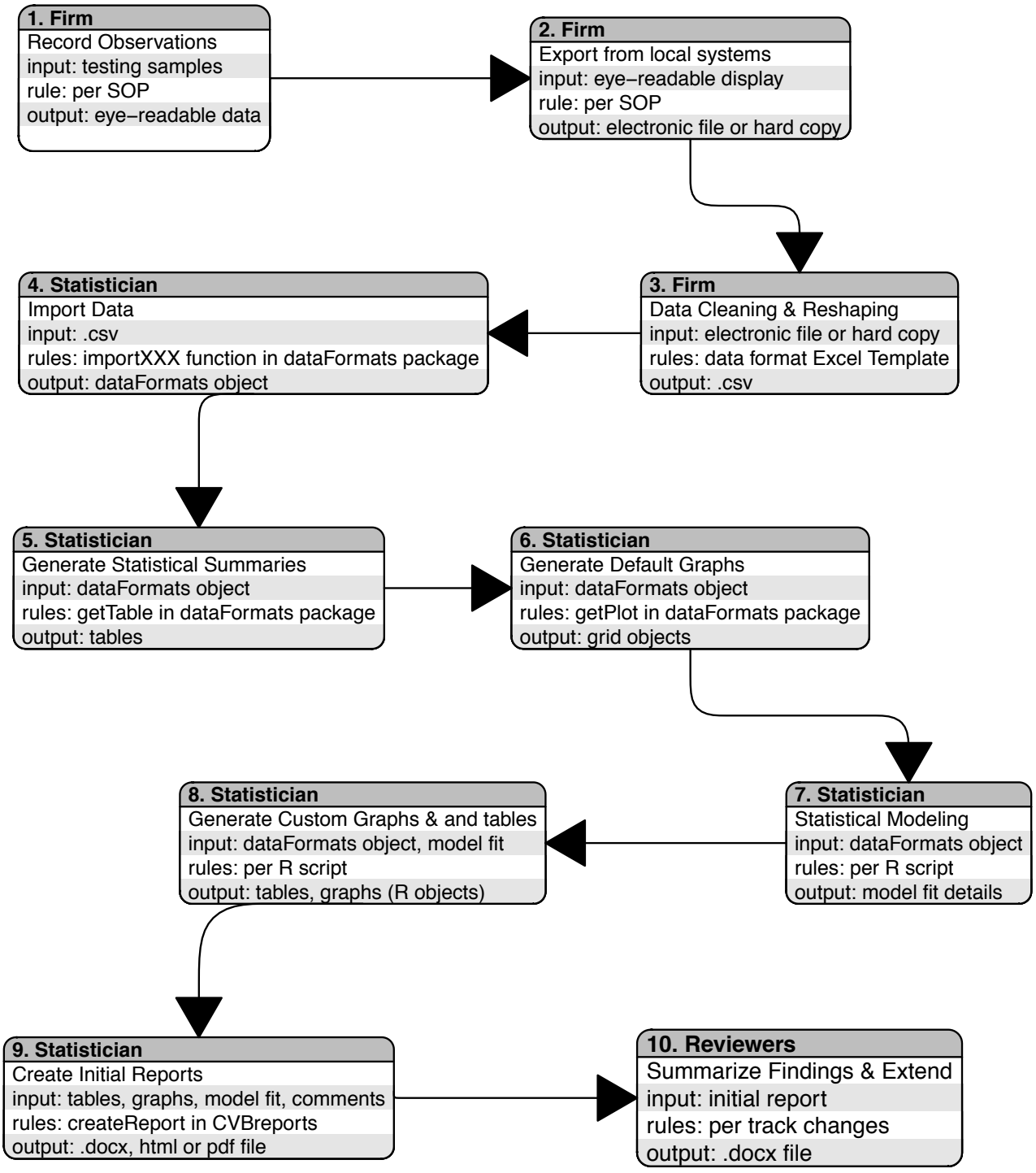
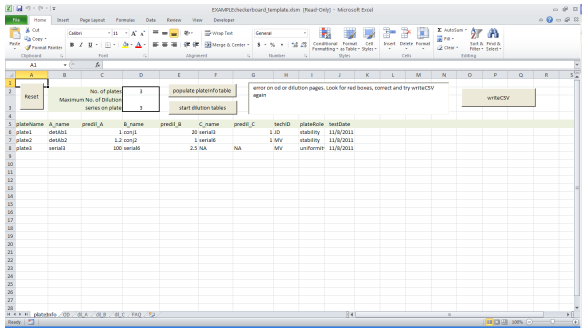
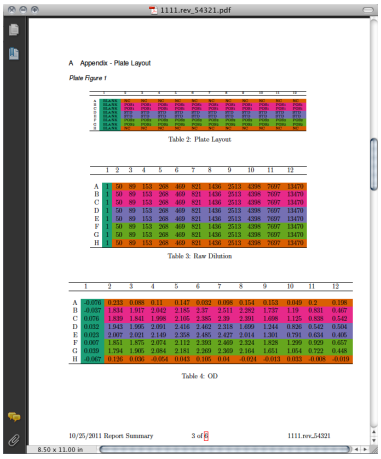


Figure 3.2: Revised CVB Statistics workflow. By developing functions for stages 4 - 6 and 8 (`importXXX`, `getTable`, `getPlot`, `createReport`), the statisticians' activity becomes a decision-making activity of selecting parameters instead of coding (including testing) and documentation. Likewise, **PF** and internally shared packages reduce burden of implementation and enable consistency across multiple statisticians.



(a) Sample of data entry. Embedded buttons and dialog box provides user with instructions as-needed and quick access to macros.



(b) A page from the sample report output. This page shows the dilution, od values and well contents for a 96-well format plate. Wells of the same color act in the same role under the experimental design (e.g. positive control, blank, serial)

Figure 3.3: Screenshots of the data entry GUI and output report. Reports may be output as html, pdf or docx files.

CHAPTER 4. VALIDATING METABOLITE DISCOVERY DURING ANALYSIS OF GC-MS DATA

A paper in submission to Metabolomics Journal

This paper originated as a collaborative term project in `STAT 503` (Instructor: D. Cook) with classmates Mahbubul Majumder (Statistics) and Tengfei Yin (MDCB) in Fall 2011. Dr. Suh-Yeon Choi (GDCB) provided the data. In collaboration with Dr. Hofmann I revised the paper, performing the literature review and extending analysis to include the purity analysis. The suggestion that cutting the hierarchial tree at a higher level to elucidate higher order descriptors is mine alone.

Abstract

A non-trivial problem when analyzing gas-chromatography mass spectrometry (GC-MS) data in the field of metabolomics is identifying relevant peaks for an entire experiment. Grouping components across multiple spectra is complex due to the variations inherent in biological systems, bench sample preparation, and observation method. We present an analytical method for examining an entire experiment for the purpose of identifying candidate metabolites for further analysis. This method enables grouping of metabolites at different levels of structural similarity across multiple spectra in an experiment.

4.1 Introduction

Traditional application of gas-chromatography mass spectrometry (GC-MS) data analysis seeks to confirm the presence and/or determine abundance of particular molecules in a bench

sample [1]. For identification, spectra are matched to a multi-peak line chart of a known standard such as in Figure 4.1. The amount of total concentration or a spectrum, is determined by scaling the observed intensity (the area under the curve of a peak) to an internal standard of known concentration. During GC-MS, bench samples are vaporized and ionized into charged particle fragments which are then accelerated through a detection chamber. The instrument reports both the derived mass-to-charge ratio (m/z, calculated from time for particles to travel a known distance) and the abundance at each or these ratios. For each molecule, there exists a unique 'finger print' in form of the distribution of relative abundance at specific m/z values. This distribution has multiple peaks due to isoforms, ionization, and fragmentation.

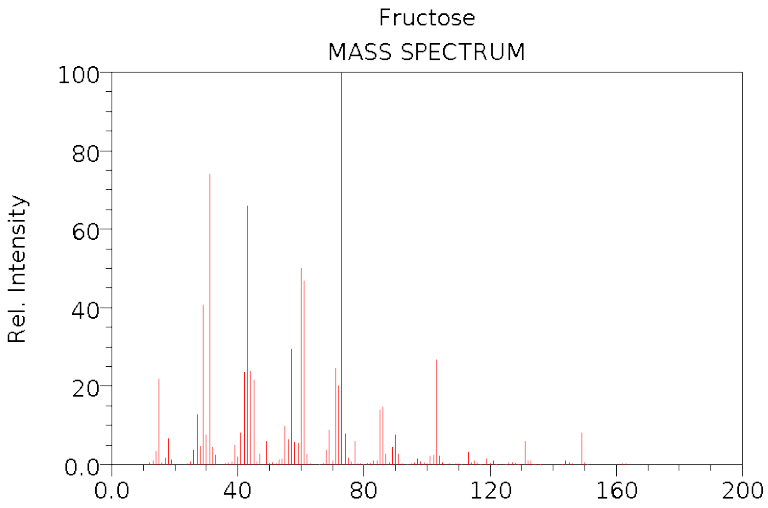


Figure 4.1: Standard spectrum for Fructose

The ability to identify chemicals present in a bench sample makes GC-MS a technique highly suited for the field of metabolomics. One of the primary goals of metabolomics is to shed light on biochemical pathways that are activated due to specific environmental conditions, during developmental stages, and/or genetic variations. In a bench sample this can be done by observing what chemicals (metabolites) are present. Unlike historical applications of GC-MS where the analysis involves identification and quantification of a few anticipated chemicals, analysis in the field of metabolomics seeks to simultaneously identify hundreds of metabolites with limited a-priori knowledge of each metabolite's identity. This vastly increases the complex-

[1] We use 'bench sample' in this text to distinguish from statistical sampling

ity of analyzing GC-MS data: first by retaining the fragmentation pattern as an unknown and second by asking researchers to simultaneously evaluate an unknown number of metabolites.

The purpose of this paper is to show that, despite the introduction of a large amount of unknowns, it is possible to perform analysis of GC-MS data in this new setting in order to generate a sensible list leading to identification of metabolites. We conducted a controlled experiment demonstrating use of clustering algorithms to identify metabolites.

We first introduce the *GC-MS Data Protocol*, which describes the steps necessary for processing data from the raw GC-MS output to a metabolomic analysis. Each step within this protocol is supported by multiple tools. A more general description of it can be found in Katajamaa and Oresic (2007) and Lawrence (2006):

1. **Baseline Subtraction.** Each method listed is an alternate method for subtracting non-informative trends, which are introduced by chromatogram testing system due to mobile phase (e.g. solvent) elements. There are a multitude of methods described in the literature: baseline subtraction can be done by using linear (Stein, 1999) or non-linear regression (Johnson et al., 2003; Baran et al., 2006), repetitive fit to a weighted loess model, running window quantile filter (Sauve and Speed, 2004; Kohlbacher et al., 2007), Jr and Jorgenson (1993); Li et al. (2005), taking the negative second derivative after smoothing by matched filter (Danielsson et al., 2002; Andreev et al., 2003; Smith et al., 2006), comparison to ions in control scans (Ruckstuhl et al., 2001; Wang et al., 2003; Fiehn et al., 2008; Kind and Fiehn, 2010).

2. **Peak Detection.** Identify individual peaks within a metabolite distribution by first estimating noise by the overall mean intensity (including baseline), the median absolute deviation of flat regions from mean, or using a moving-window fit to smooth (Stein, 1999; Smith et al., 2006; Morris et al., 2005; Hastings et al., 2002). Next, estimate the region of x-axis occupied by a peak using a fixed size window (Stein, 1999) or zero-crossings (Christensen et al., 2005; Smith et al., 2006). Finally, fit a model to the peak (Stein, 1999; Du et al., 2006; Pluskal et al., 2010; Lommen, 2009; Lan and Jorgenson, 2001; Christensen et al., 2005).

3. **Component Detection.** Group multiple individual peaks into the distribution pattern for a single metabolite, using either binning (Stein, 1999) or overlapping peak volume (Ahmad et al., 2011)

4. **Component Grouping across spectra.** Match metabolites across multiple bench samples in the experiment using various clustering (Johnson et al., 2003; Katajamaa and Oresic, 2007; Bellew et al., 2006) or model-based clustering (Smith et al., 2006; Wang et al., 2007; Jaffe et al., 2006; Yu et al., 2006; Rogers et al., 2011)

5. **Retention Time Correction.** Correct for retention time drift between runs on an instrument using correlation optimized warping Nielsen et al. (1998); Baran et al. (2006), Hidden Markov models (Listgarten et al., 2005), parametric time warping (Jaffe et al., 2006; Eilers, 2004; Kohlbacher et al., 2007), or distance functions (Hoffmann and Stoye, 2009) .

6. **Summarization.** Calculate area under the metabolite peaks.

7. **Normalization.** Normalize for differing metabolite levels between spectra due to error.

8. **Metabolite identification.** Compare to control spectra and database listings. This step is not automated and may refer to proprietary databases such as the NIST Mass Spectral Library (Stein, 1999) and through Pacific Northwest National Laboratories (Sharpe et al., 2004)

The output from the GC-MS Data Protocol consists of a list of metabolites that are in each spectrum. For each metabolite, this analysis also provides the retention time, mass ratio (m/z) and abundance (standardized across all spectra in the experiment) of individual peaks.

Each stage of the workflow includes choices in methodology and parameters for the data processing. This introduces additional variability. It is therefore imperative that we validate the results of processing using independent and statistically robust methodology.

In this paper, we validate GC-MS Data Protocol by applying clustering techniques to the output from step 3 above and compare results to experimental design of a validation experiment.

In the next section we describe the experimental design and expected data outcomes. Section 3 identifies pitfalls of applying traditional mass spectrometry analysis to metabolomics data without modification.

4.2 Experimental Design

In order to validate the GC-MS Data Protocol, we have designed a fully controlled experiment intended to mimic a typical metabolomics experiment: two unique metabolite mixtures (A and B) represent different experimental conditions (e.g. wild-type and growth condition). Each of the mixtures consists of a set of (artificial) metabolites, some of which include metabolites that are unique to that group and others that are common to both. Mixtures were prepared twice (experimental replicates) and two aliquots from each replicate were run (technical replicates), for a total of 4 replicates for each mixture. Included are multiple metabolite classes (amino acids, sugars, sugar alcohols, and various organic acids) with a wide range of retention times. Table 4.1 summarizes the metabolites in each mixture, including class and theoretical retention times.

The expectation is that, upon analysis using the GC-MS Data Protocol, one metabolite record is generated per chemical entity included in each bench sample. Additional expectations include: (1) records representing the same metabolite in different samples demonstrate greater similarity than distinct metabolites (2) unique metabolites show similarity that correlates to differences in structure and (3) the number of metabolites and level of similarity is reflective of the underlying experimental design (e.g. The validation data would show 25 total metabolites, 9 of which have four records [two from technical replicates of mixture A and two from technical replicates of mixture B]. Records that may be traced to trehalose, maltose and sucrose would show greater similiarity than when compared to other records in the validation data set).

4.3 Limitations of a Straightforward Approach

For the analysis presented here, files generated from the spectrometer were initially processed using the `R` package `chromatoplots` (Lawrence et al., 2012b) to identify peaks that are

likely candidates to represent individual metabolites of interest. `chromatoplots` follows the GC-MS Data Protocol. For each metabolite, identified by a component identifier, we have: treatment (A or B), biological replicate 1 or 2 and technical replicate 1 or 2, the number of peaks in the component, a list of peaks with corresponding m/z ratios and raw intensities, and retention time . Additional summary details are included in the variables: standard deviation of peak intensities and standard deviation of rt.

The initial output across the entire experiment consisted of 1886 metabolite records. This is much more than the anticipated 136 records (17 metabolites per sample x 4 samples per mixture x 2 mixtures). The analysis is additionaly complicated by the fact that length of the peak list for each record varies in length. Variation is simply due to the fact that different metabolites fragment uniquely, but it means not only does the number of variables change, so does the identity of those variables.

4.4 Filtering and New Variables

The mechanics of mass spectrometry make it highly unlikely to observe a real metabolite with only one peak because simply the gain/loss of one hydrogen or water molecule is sufficient to generate a second peak. Out of the original 1886 candidate metabolites 520 rows of data were excluded due to a presence of only one peak (npeaks = 1) for that component.

A second filter is to screen for metabolites with the greatest representation, or highest cumulative intensity. This cumulative value is the sum of all intensities for all peaks in the metabolite and is a measure of the number and abundance of molecular fragments associated with a metabolite. Even trace amounts of a molecular ion will display intensities many orders of magnitudes greater than the variation that is a byproduct of GC-MS and sample preparation techniques. Filtering at the 90th quantile of cumulative intensity reduces the dataset to 137 rows.

A feature that is key to accurate chemical identification in manual processes is the unique fingerprint of fragmentation. Metabolites may be identified by the combination of m/z ratio values and the relative intensity between peaks. To retain some m/z and intensity information the m/z ratios of the two peaks of greatest and second-highest intensity, as well as the maximum

peak intensity divided by the peak intensity of the second highest peak. Finally we captured the presence of the highest peak intensity compared to the average intensity values for a component.

4.5 Methods

With multiple algorithms widely accepted (Do Yup Lee and Northen, 2010) for individual stages of the GC-MS Protocol, there is no single gold standard for the full analysis. Regardless of the choice of individual algorithms, the mass spectrum output from the spectrophotometer is a histogram showing the pattern of the distribution of the metabolite's ion fragments, a pattern which is unique to the metabolite and reproducible under consistent sample preparation and detection techniques. Furthermore, pattern similarities exist between metabolites that fragment into the the same ions. The motivation for fragmentation into the same ions lies in the energetic properties of the originating metabolite which in turn bears a direct relationship with the underlying chemical structure. Thus, two metabolites with similar chemical structure are expected to share common features in their respective spectra. Using the variables discussed earlier as a quantitative summary of a spectrum, the difference between all pairs of metabolites may be represented by a distance matrix. We apply hierarchical clustering(Ward, 1963) to make use of distance measures and combine similar metabolites (spectra) into the same group ('cluster').

Expected outcomes from Section 4.2 can be described in expanded form:

1. Replicates of same metabolite are in the same cluster. and data collection, the only observed difference between metabolite replicates is the sample identifier.

2. Distances between clusters is a measure of structural similarity. The ions generated during fragmentation come about due to energy excitation disrupting chemical bonds. Metabolites with similar energetic properties fragment with a similar pattern.

3. Cluster sizes are unequal. The diameter of a single cluster is dependent on the strength of its underlying similarity. It is not unreasonable to allow for the scenario that within-group similarity is inhomogeneous from cluster to cluster.

Figure 4.2a shows the theoretical cluster dendrogram. Reading left to right, metabolites first separate conditional on chemical class: amino acid, monosaccharide, sugar alcohol, disaccharide, organic acid (cyclic) and organic acid (branched), a superficial level of structural similarity. Without considering the impact of data collection activities, the next level of clustering would be at the metabolite level. However, given the experimental design underlying this validation data, we expect clustering to occur both due to experimental replication and at a lower level due to technical replication.

In the rest of this section, we focus on how the actual results differ from theoretical expectations.

4.5.1 Known metabolites

Clustering of the validation dataset (Figure 4.2b) and comparing to the theoretical dendrogram (Figure 4.2a) allows us to see the initial division of metabolites into groups that may be attributed to the chemical class. It is possible to cut the tree derived from experimental data such that initial branching matches the structure suggested by the tree based on theoretical information, effectively cutting the tree with a choice of $k = 6$. Closer inspection of the leaves of Figure 4.2b shows additional branching not initially predicted. Additional branching may suggest that peaks that are due to excessive noise or other peaks unresolved by the chromatoplots algorithm. Differences between trees calculated using experimental information versus theoretical data suggest the need for refinement of parameter choices passed to chromatoplots. The impact of chromatoplots parameter selection is further discussed (ref). The choice of $k = 6$ requires knowing general information regarding metabolite identity, which is not realistic. Researchers working with datasets of unknown composition do not have the luxury of knowing a priori what chemical classes are in the samples.

This suggests that seeking to identify a value for k by closely examining the possible biochemical structures may yield a selection that is uninformative regarding metabolite identity. Furthermore, in an experimental setting, discovering all structural classes that are in particular sample is labor-intensive, and possibly idiosyncratic. For clustering to be a viable analytical approach, we need a practical strategy for determining k.

4.5.2 Identifying k from graphs - a brute force approach

Regardless of experimental design or *a priori* knowledge, the pedigree of a particular leaf of a cluster is a known value. The pedigree of a single leaf is simply an identifier associated with the originating electronic file (i.e. output from mass spectrometer) - and by extension the originating sample. Also known are the number samples which are replicates (either technical or experimental). Because the expected tree is such that replicates are the tightest cluster, an investigator can identify a likely value for k by examining all trees produced by varying k from one (1) to n, where n = number of leaves. The tree that shows 'pure' clusters - all leaves from the same replicate group - with a member size that matches the number of replicates as designated by the experimental design is the one that demonstrates a likely choice for k. An illustration of this examination process is shown in Figure 4.3, for a subset of values for k - a complete examination of the data in this case study would include plots for 137 choices of k. Using histograms of the number of metabolites assigned to each cluster, factored by mixture, is a graphical approach to examination.

While this method neatly sidesteps the issue of knowing the number of metabolites present in a bench sample, it also presents a computational challenge in practical application. Bench samples (e.g. leaf extract) may contain hundreds of metabolites. Identifying a value for k would require computing and evaluating a hierarchical tree for each.

4.5.3 Determining k using probabilities

We propose the following method for identifying a value for k which is independent of a priori knowledge.

From the *compID* of each metabolite peak within a cluster, a new variable can be created for the cluster — purity — which is simply a binary indicator whether the cluster is pure (all members are from the same mixture) or not (members are from more than one mixture). For each cluster, we examine the proportion of observations from each treatment, which we will denote by p_A and p_B. A cluster is *pure*, if $p_A = 1$ or $p_B = 1$. Purity of a clustering result can be summarized – for a specific value of k – as the proportion of clusters are pure.

The probability of node c of size n being composed of a single mixture is:

$$P_{pure}(c) = p_A^n \cdot p_B^0 + p_B^0 \cdot p_B^n \quad (1)$$

$$\sum_{i=1}^{k} P(\text{pure } c_i) = \sum_{i=1}^{k} (p_A^{n_i} + p_B^{n_i}) \quad (2)$$

In our example we know that the theoretical probability of a metabolite in a particular mixture should be $p_A = p_B = 0.5$, which can be used in equation (2) to calculate the probability of purity.

It is unlikely in an experimental setting to have foreknowledge regarding the distribution of unique metabolites across sample preparations (e.g. A or B). However because we know which spectrum (data file) an observed metabolite comes from, and the sample preparation (A or B) which was processed to generate that spectrum, we can calculate an empirical probability $\hat{p}_A$ and $\hat{p}_B$ from the data directly. In this case the probability that an observed metabolite (one record in the data table) is from a particular treatment (mixture) is:

$$\hat{p}_A = \frac{69}{137}, \text{ and } \hat{p}_B = \frac{68}{137}$$

Solving for equation (2), this leads to the theoretical probabilities of Figure 4.4a. Standardizing the purity probability by k allows for examination of all possibilities of k. The choice to normalize by k reflects the expectation that as k increases, the probability for a pure node also increases, i.e. for $k \to \infty$, P(pure) $\to$1.

To find choices for k which are of greatest interest, we compare the empircal probablity of node purity with expected purity probabilities calculated as above. This purity ratio yields a curve with a strong decay for large values as shown in Figure 4.4a. Values for k which bear further investigation correlate to positions on plotted curve which exhibit local jumps when deviating from the expected approach to an asymptote of 1. Jumps indicate that a particular choice of k is more informative than expected.

From Figure 4.4b, we see that there four points with jumps greater than 0.1. These observations correspond to values for k of 6, 10, 12, 25. For this validation case, we know that 6 is the number of classes by chemical structure. A k of 25 corresponds to the total number of unique metabolites present.

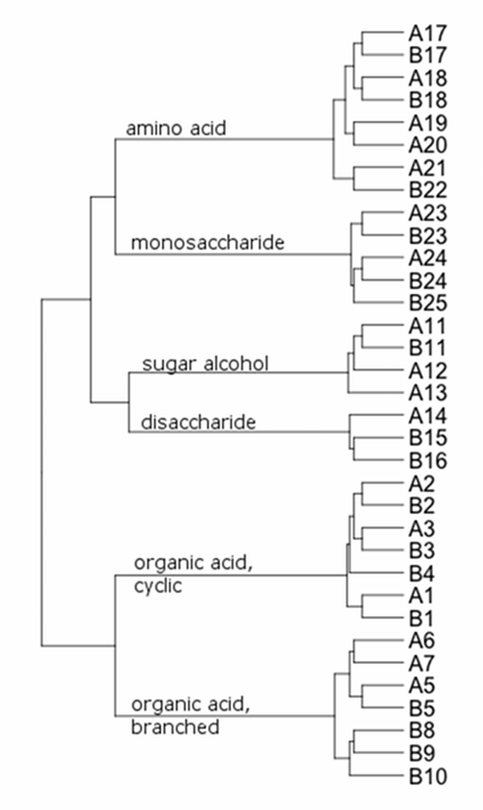
With a k of 25, we see that the presence of metabolites by treatment (mixture id) presents researchers with an overview of the overall experimental design. Figure 4.5 indicates that approximately 1/3 of the metabolites (clusters) are only present in Mixture A, 1/3 in Mixture B and 1/3 in both mixtures. This is the anticipated distribution per the sample contents listed in Table 4.1

4.6 Conclusion

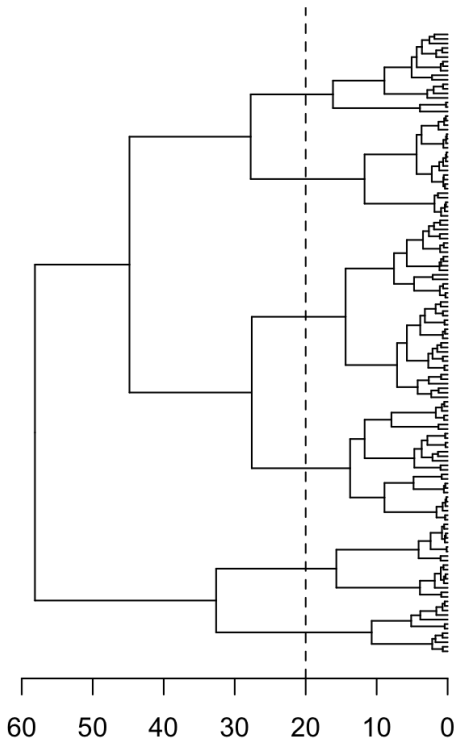
The analysis presented above validates a workflow for elucidating metabolite distribution among samples of a typical metabolomics experiment using clustering methods to extend the commonly accepted *GC-MS Data Protocol*. We started with data output applying the GC-MS Data protocol to an experiment of known design and metabolite composition. We filtered data records from 1886 to 137 entries using criteria defined by the mechanics of mass spectrometry. For these records we created new quantitative variables that describe unique fragmentation pattern of a metabolite. We applied hierarchical clustering to group records by metabolite identity. To determine a value for k in the clustering algorithm, we defined a new variable, the *Purity Ratio*, which is a measure comparing the observed purity of one cluster compare to the purity that suggested by the experimental design. The purity calculations are a technique for identifying interesting values for k that are of scientific interest and may be associated either with structural classes of metabolites or individual metabolites. With empirically determined choices for the number of clusters to use in hierarchical clustering we determine identify the experimental design without additional a priori knowledge. When applying this approach to the validation data set, we see a distribution between pure and mixed clusters that are consistent with the underlying metabolite composition.

Class	Metabolite	Conc (mg/ml)	Volume (ml)	Retention Time (min)	Mixture A	Mixture B	Label
organic acid	shikimic acid	10	20	24.6	■	■	1
organic acid	pimaric acid	1	40	19.64	■	■	2
organic acid	isocitric acid	10	20	24.81	■	■	3
organic acid	quinic acid	10	20	25.62		■	4
organic acid	adipic acid	1	40	17.39	■	■	5
organic acid	citric acid	10	20	24.69	■		6
organic acid	malic acid	10	20	17.28	■		7
organic acid	succinic acid	10	20	12.54		■	8
organic acid	malonic acid	10	20	9.63		■	9
organic acid	keto-isovaleric acid	10.1	20	6.99, 7.72		■	10
sugar alcohol	erythritol	10	20	17.98	■	■	11
sugar alcohol	arabitol	10	20	22.79	■		12
sugar alcohol	mannitol	10	20	27	■		13
disaccharide	trehalose	10	20	40.5	■		14
disaccharide	maltose	10	20	40.53, 40.87 (+40.341)		■	15
disaccharide	sucrose	10	20	39.1		■	16
amino acid	lysine	13.1	20	26.47	■	■	17
animo acid	3-amino-butyrate	4.4	20	9.69	■	■	18
amino acid	glycine	15.3	20	12.32	■		19
amino acid	leucine	9.6	20	11.49	■		20
amino acid	4-amino-butyrate	2.8	20	17.92	■		21
amino acid	glutamine	17.9	20	17.67		■	22
monosaccharide	glucose	1	40	26.45 (+26.71)	■	■	23
monosaccharide	fructose	1	40	25.95, 26.18	■	■	24
monosaccharide	ribose	10	20	21.83		■	25

Table 4.1: Metabolites contained in each mixture. Mixture A contains 17 substances, mixture B contains 17 substances. Nine substances exist in both mixtures. Theoretical retention times are also provided.



(a) Expected hierarchical tree, cut to show clusters by metabolite structural class. Leaves are labelled with both mixture ID (A or B) and compID (row number from Table). It is anticipated that, under experimental conditions, the leaves shown would demonstrate further branching with presence of replicates.



(b) Hierarchical clustering of experimental data using Ward's method

Figure 4.2: Theoretical and observed hierarchical trees

class	Mixture A	Mixture B	Both	Total
organic acid (cyclic)	0	1	3	4
organic acid (branched)	2	3	1	6
sugar alcohol	2	0	1	3
disaccharide	1	2	0	3
amino acid	3	1	2	6
monosacharide	0	1	2	3

Table 4.2: Expected proportion of metabolites by chemical class

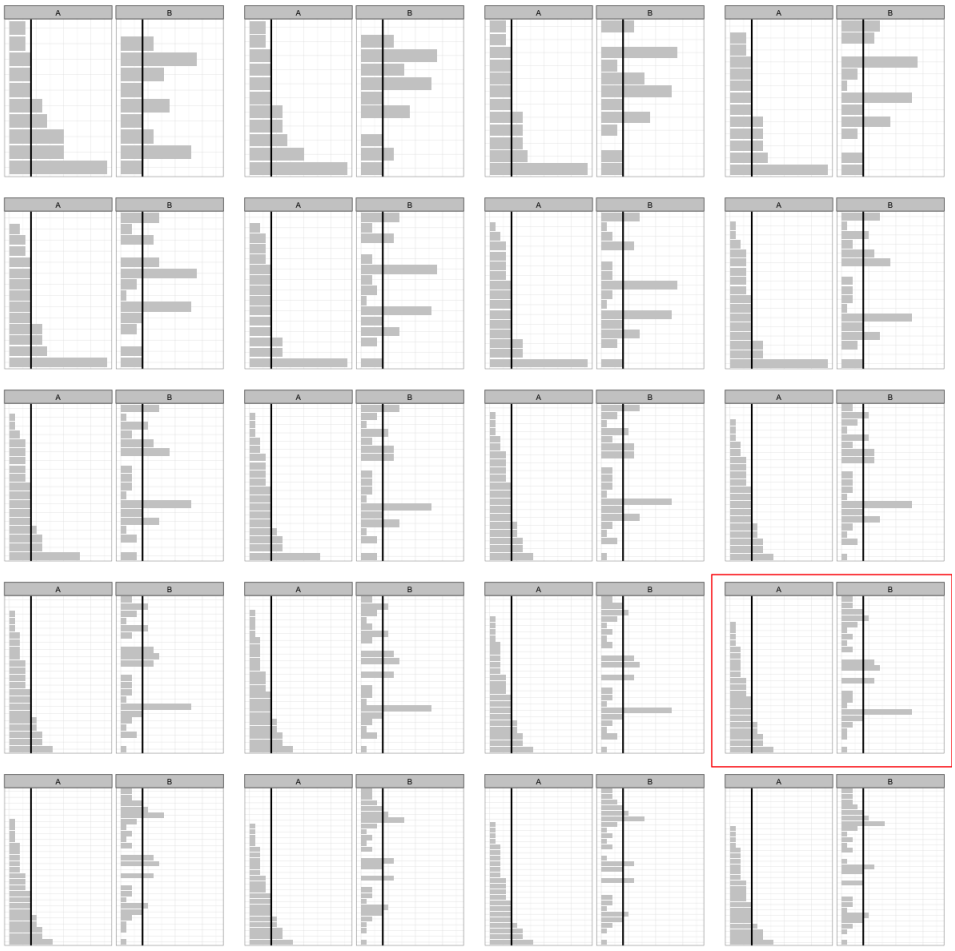
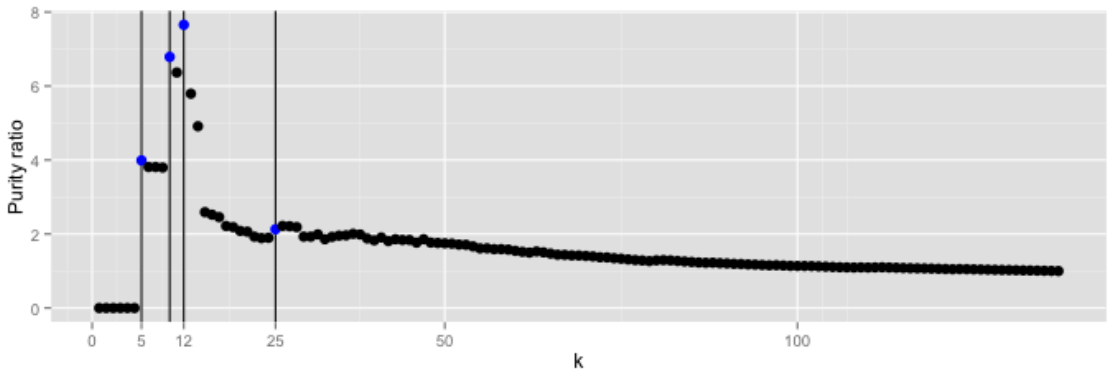
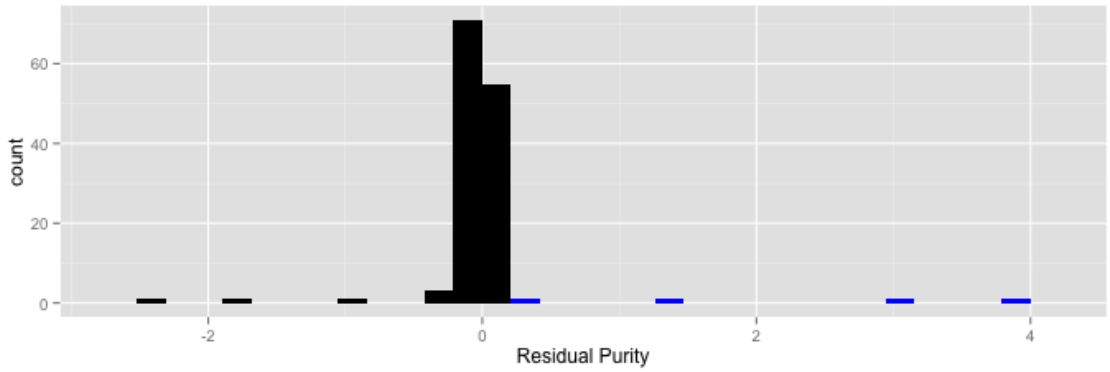


Figure 4.3: Comparing the purity of clusters by choice of k for $k = 10, 11, ..., 29$. For each choice of k, a tree is created (not shown, example in Figure 4.2b). The number of metabolites per cluster is shown as a histogram with each bar as a metabolite cluster, conditioned by mixture. Highlighted in the red box is the plot for k = 25. Each plot also show guides for count = 4, the number of replicates for each mixture (total of experimental and technical). Given the experimental design of this validation study, correct clustering is demonstrated by eight clusters (bars) present in mixture A only with a count of four, eight clusters (bars) present in mixture B only with a count of four and nine clusters with a count of four each in mixtures A and B. A complete analysis of all possible choices would involve examining plots for all possible values for k (1 to 137) for presence of metabolite clusters that are purely mixture A, purely mixture B or an equal combination of both, but without bias towards the number of clusters.



(a) Ratio of observed cluster purity compared to expected cluster purity. Jumps that deviate from the overall decay curve are choices for k that bear further interest.



(b) The difference between purity ratios for k vs $k-1$ indicate that of the 137 potential values for k, only four choices (6, 10, 12, 25) are of high interest.

Figure 4.4: Identifying k from the difference of purity ratios.

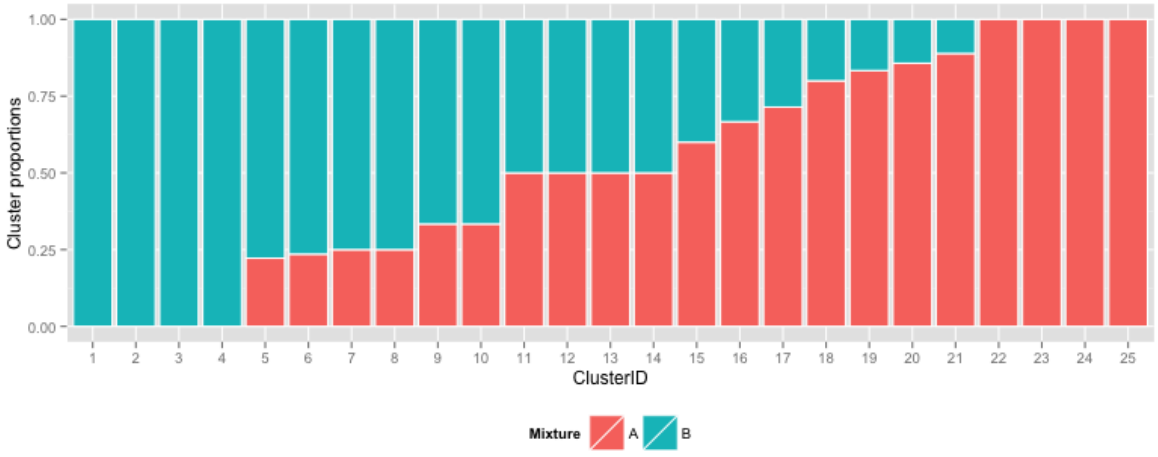


Figure 4.5: Cluster proportions when k = 25. We see that, as expected, some clusters are all mixture A, some are entirely comprised of mixture B and some consist of equal membership mixture A and mixture B.

CHAPTER 5. TOOLS: CHROMATOPLOTS GUI

One weakness in tools for biological analyses in the field of metabolomics is the absence of options that leverage robust visualizations (i.e. incorporating concepts from human computer interaction within a reproducible environment (e.g. how to capture user settings in Chemstation software?). **chromatoplotsGUI** is a set of interactive visualizations for each stage within the chromatoplots workflow, organized into a graphical user interface. While the package chromatoplots captures the analysis values in the pipeline framework and records the necessary values to recreate the analysis steps, it does not provide opportunity to extend pipeline management to information captured via an interactive graphic. An additional step needed to finalize development of chromatoplots is validation using a known data set. For the chromatoplotsGUI interface, there are two sections: (1) creating functions to independently generate interactive graphics and (2) compilation of these interactive graphics into a single graphical user interface. The draft text for validation is in Chapter 4 and the code for **chromatoplotsGUI** may be found online at: https://github.com/mariev/chromatoplotsgui. **chromatoplotsGUI** is written using **cranvas** plots for data visualizations, which may be accessed directly by command-line or via a user interface viewed using an internet browser.

5.1 Debugging chromatoplots

When initially developing **chromatoplotsGUI**, **chromatoplots**, the underlying computational engine for data processing unique to the workflow described in detail in Chapter 4, was also in an early stage of development. This presented an opportunity to use interactive graphical tools for debugging the underlying algorithms of **chromatoplots**.

After loading data the next stage of processing is to organize data in to matrix (tabular)

form, with rows and columns representing binned values for time and m/z ratio respectively. A single cell contains the intensity observed for a particular binning of time and m/z. At intial data collection, these many of these cells do not contain observed values of intensity. To prepare for later stages of analysis, namely curve fitting, it is necessary to popluate the empty cells with a default value. In **chromatoplots** this is done using the `genProfile` method. For one spectrum from the experiment described in Chapter 4, applying this method increases the data matrix from 190,288 entries to 5,048,250 profile points. *What was undocumented at the time was that the default value for these auto-populated positions deviated from established defaults.* That is, functions to perform this step in other R packages (Smith et al., 2006; Tautenhahn et al., 2008; Benton et al., 2010) use the default value of $0.5 * min(intensity)$. **chromatoplots** implemented a default of `min(intensity)`. This generated a resultant data matrix wherein a researcher could not distinguish between data points observed at minimum values (*true minimums*) and points labeled with minimum intensity values due to an artifact of computation. True minimums represent a set of *censored data* - cases when the mass spectrometer was able to detect presence of a metabolite (intensity) but at levels below the instrument's sensitivity threshold. In the spectrum described, this loss of information was affects only 1,453 data points out of the original 190,288 values (0.76%). The inconsistency and loss of information affecting such a small subset of data was only detected during the development of **chromatoplotGUI** interactive plots and may be attributed to (a) the ability to quickly zoom into an area of visual interest and (b) the speed with which plots are rendered. While the graphics rendering in **chromatoplotsGUI**, which depends on **cranvas** can plot large data sets, the overhead of background analysis on over 5 million points causes an observable lag to users so artifical minimums are not displayed, neccesitating the activity of identifying true minimums. In Figure 5.6 we see the impact the choice of default values makes. When the default is set so that true minimums are indistinguishable, those values are lost during display (left). If a default choice is unique, it is possible to create displays that show points at instrument minimus (black points, right).

5.2 Linked Plots

By far the greatest control of default visualizations is via usage at the command line. As analysts step through each stage of analysis in R, they may display the data using commands to call plotting of data object generated using functions of packages **xcms**, **chromatoplots** or custom methods as desired. The commands to call visualizations are:

`cgRawPlot` A single plot that displays raw data. Time is shown on the x-axis, m/z on the y-axis, and log(intensity) is indicated by color (Figure 5.2).

`cgProfPlot` A single plot that displays raw data. Time is shown on the x-axis, m/z on the y-axis, and log(intensity) is indicated by color (Figure 5.3).

`cgRmBasePlot` A set of two linked plots. One is as for `cgProfPlot`. The second is an intensity curve for a specific m/z choice. Choice of m/z is indicated with the horizontal red line in the first plot and noted in the header for the second plot. Users can update the m/z choice by selecting a value (mouse-click) in the first plot (Figure 5.4).

`cgfindPeaksPlot` A set of plots showing (1)`cgProfPlot` output overlaid with the output of the `findPeaks` stage. Selector for the m/z ratio used in the second plot is indicated by the horizontal red line, which may be updated by mouse-clicking on the first plot to select a new value. (2) For the selected m/z, an intensity curve as in figure 5.4(right) is shown with the `findPeaks` output overlaid. Pressing "z" while this plot is active opens (3) a plot showing the "h" coefficient of the Exponential-Gaussian hybrid, as implemented in `egh` function in **chromatoplots** for each peak determined using `findPeaks` (Figure 5.5).

5.3 Using the GUI

Graphical user interfaces exist to address two issues: (1) unfamiliarity that is common with novice and infrequent users and (2) cognitive burden associated with coordination of multi-stage and complex tasks. The interface for **chromatoplotsgui** was written using the R package **shiny** and is viewed through an internet browser. From the navigation bar, users are initially presented with a "Quick-start" guide and resources to seek additonal documentation. For

analysis, there are the options of "New" or "Resume". "New" allows users to load raw `.CDF` files as output from a GC-MS instrument to work through the entire workflow as described in Chapter 4 and save data outputs at any point. "Resume" allows users to load a file previously saved by **chromatoplotsGUI** and continue analysis. Examples of the GUI are in Figures 5.7, 5.8, 5.9, 5.10, 5.11.

5.4 Implementation

chromatoplotsGUI was written in `R` v2.15.2 (Trick-orTreat) and depends on **chromatoplots** v0.0.9 (downloaded from `https://github.com/tengfei/chromatoplots` March 2013, most recent version available). Other non-standard dependencies include: **cranvas** and **shinyIncubator**. The user interface is optimized for use in chrome, v20.0. The package may be downloaded at: `https://github.com/mariev/chromatoplotsgui`

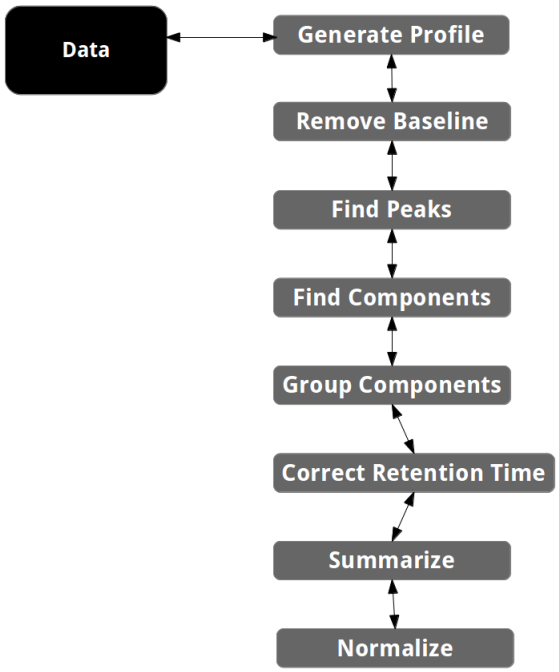


Figure 5.1: Steps of the chromatoplots workflow

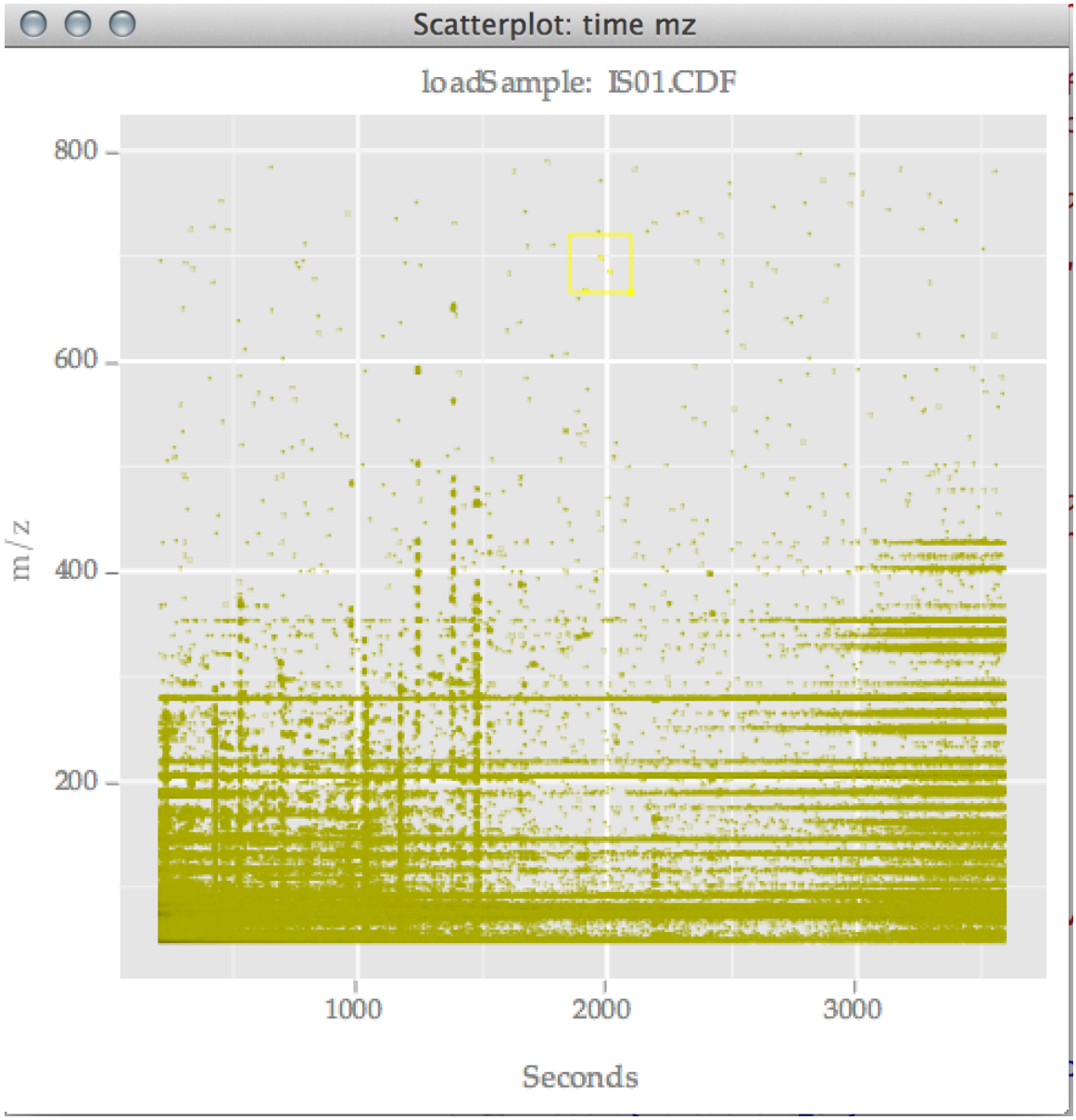


Figure 5.2: Visualization of the raw data. Interactive elements include: panning, zooming, brushing (highlighting), changing size of data points

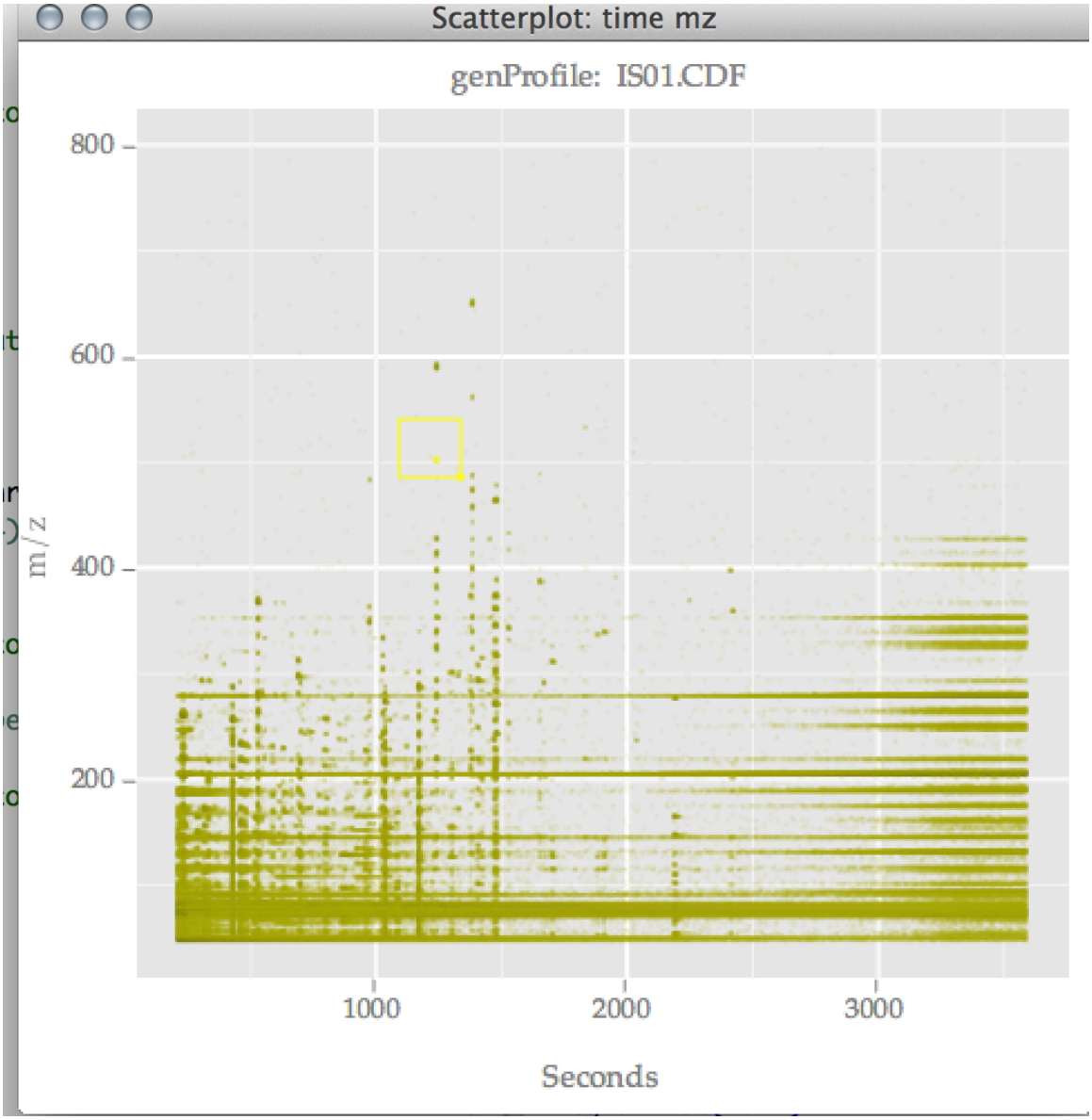


Figure 5.3: Data after `genProfile` stage. Interactive elements include: zooming, brushing (highlighting), changing size of data points

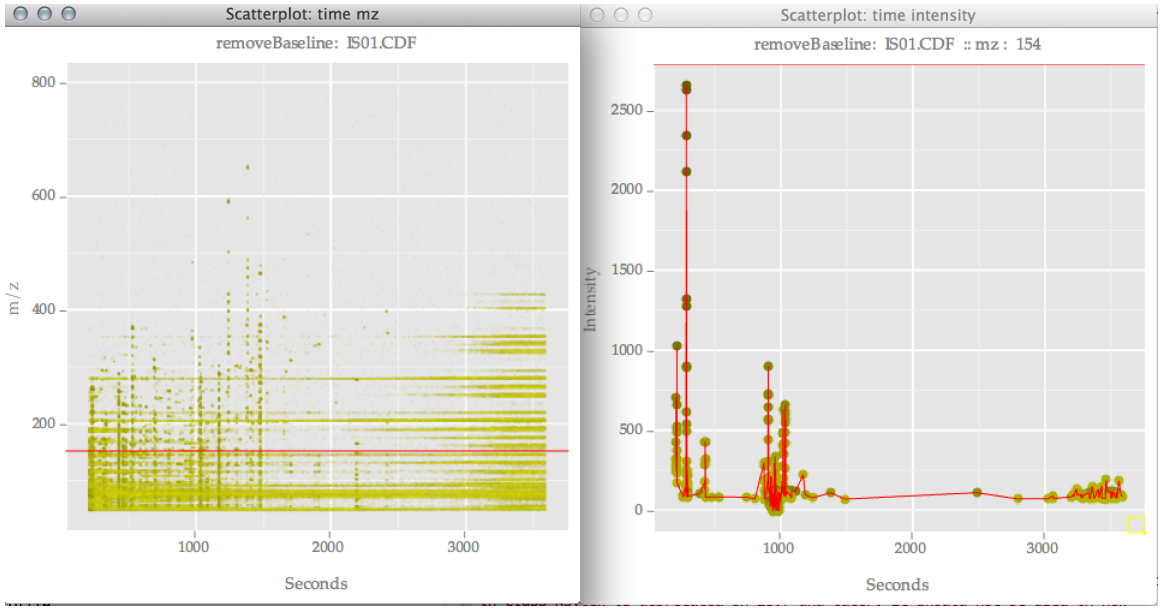
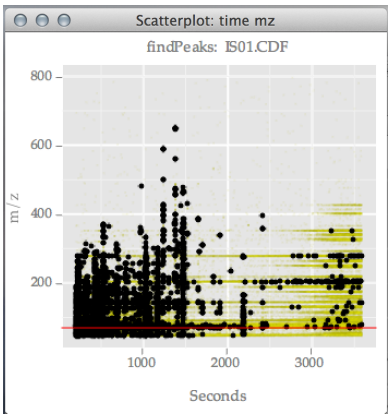
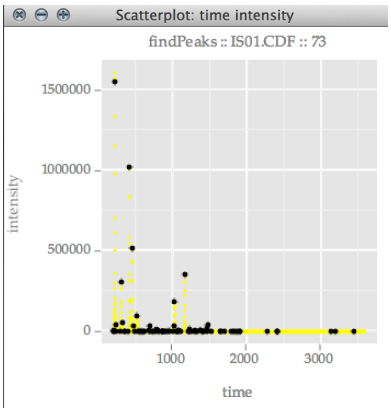


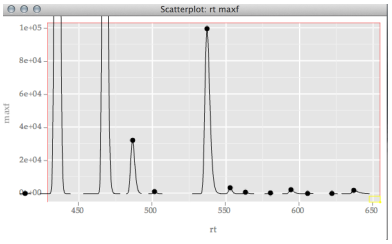
Figure 5.4: Linked plots after `removeBaseline`. (left) As for `genProfile` (right) Intensity curve for a specific m/z choice. Choice of m/z is indicated with the horizontal red line in (left) and noted in the header for the graph at (right). Users can update the m/z choice by selecting a value (mouse-click) on the left-hand plot. Both plots have interactivity of: zooming and changing size of data points, which operate independently.



(a) Plot shows the `genProfile` (yellow points) output overlaid with the output of the `findPeaks` stage. Selector for the m/z ratio used in figure 5.5b indicated by the horizontal red line. Click on the plot to update the m/z choice. Independent interactivity include: zooming and changing size of data points.



(b) For the selected m/z, an intensity curve as in figure 5.4(right) is shown with the `findPeaks` output overlaid. The plot has independent interactivity of highlighting, zooming and changing data point size. Pressing "z" while this plot is active opens figure 5.5c.



(c) Plot showing the "h" coefficient of the Exponential-Gaussian hybrid, as implemented in `egh` function in **chromatoplots** for each peak found in the `findPeaks`

Figure 5.5: Visualizing the `findPeaks` output includes three linked plots.

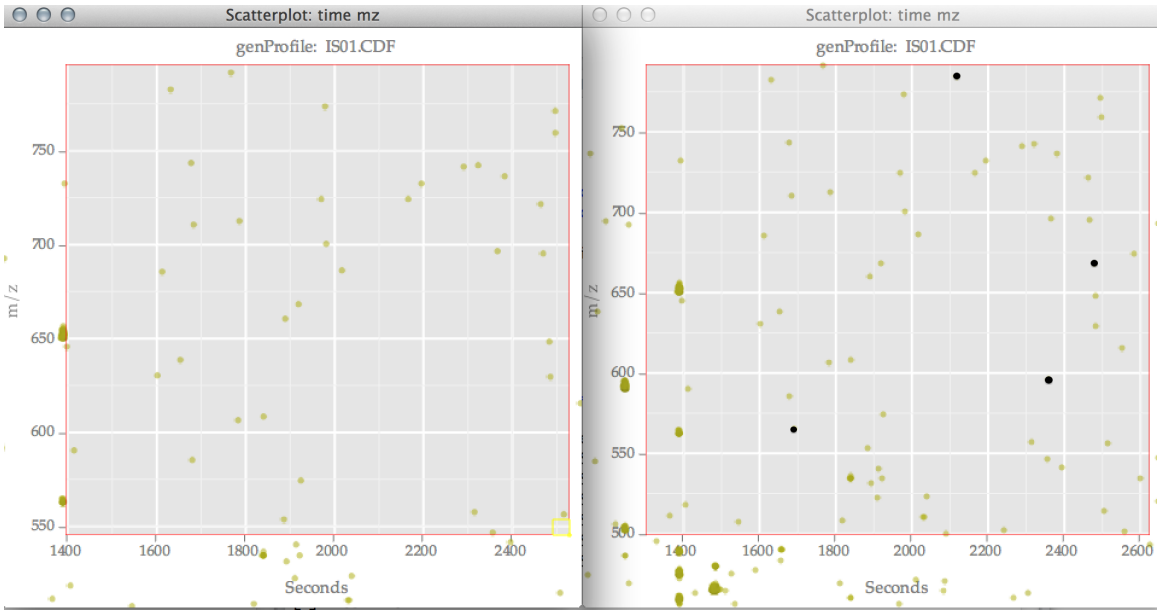


Figure 5.6: Plots for the same data spectrum (left) using original **chromatoplots** defaults and (right) using defaults consistent with **xcms** after the `genProfile` stage. Points marked in black are present when the updated thresholds are used, but were lost in original implementation.

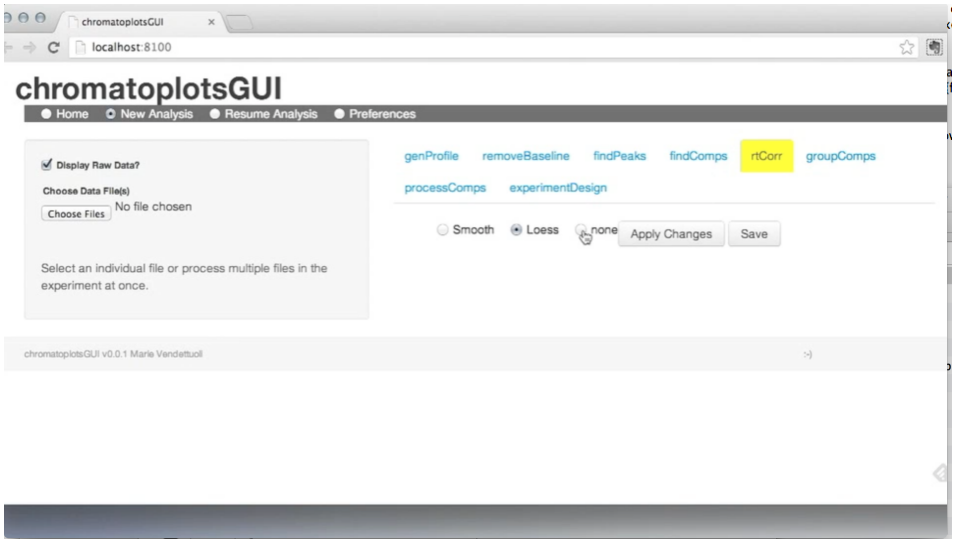


Figure 5.7: Options for "New" analysis

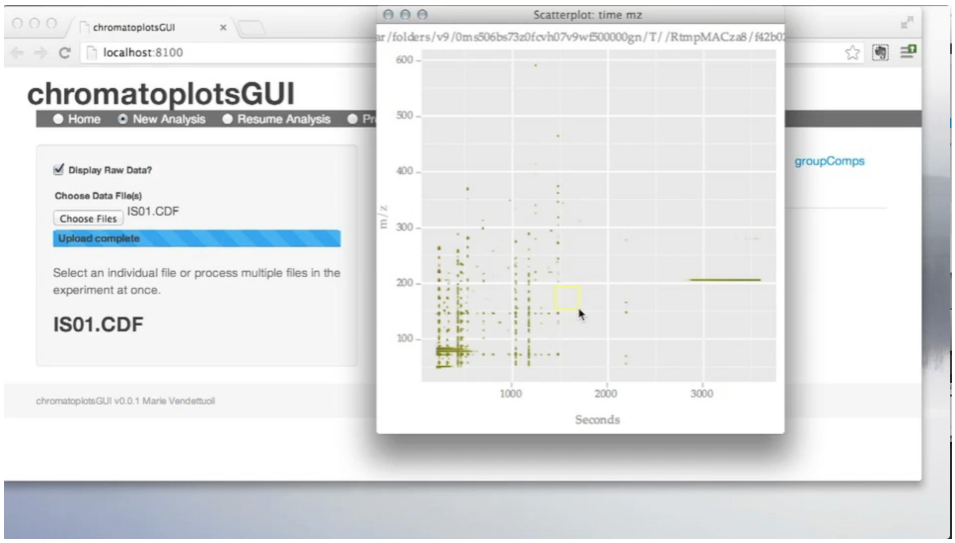


Figure 5.8: Using **chromatoplotsGUI** to open a single spectrum

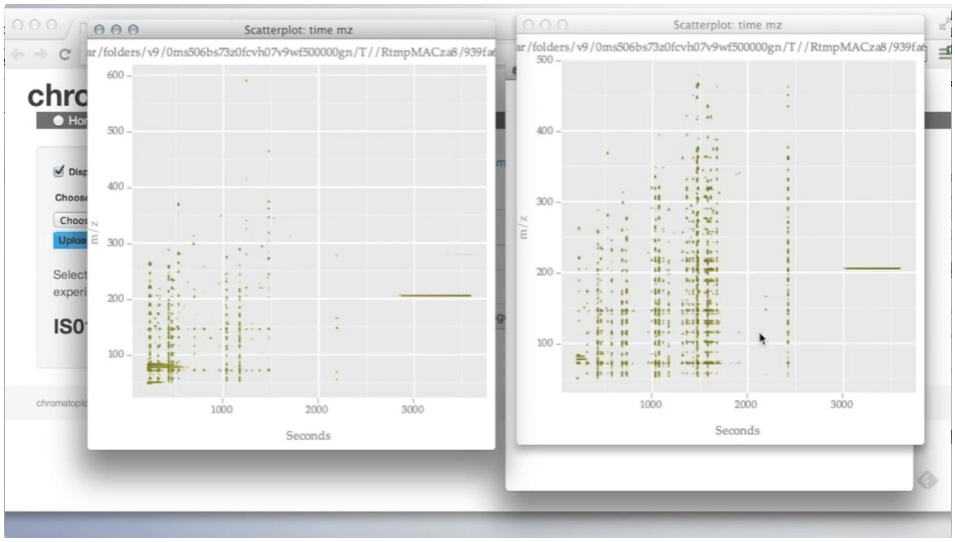


Figure 5.9: Opening multiple spectra simultaneously with **chromatoplotsGUI**

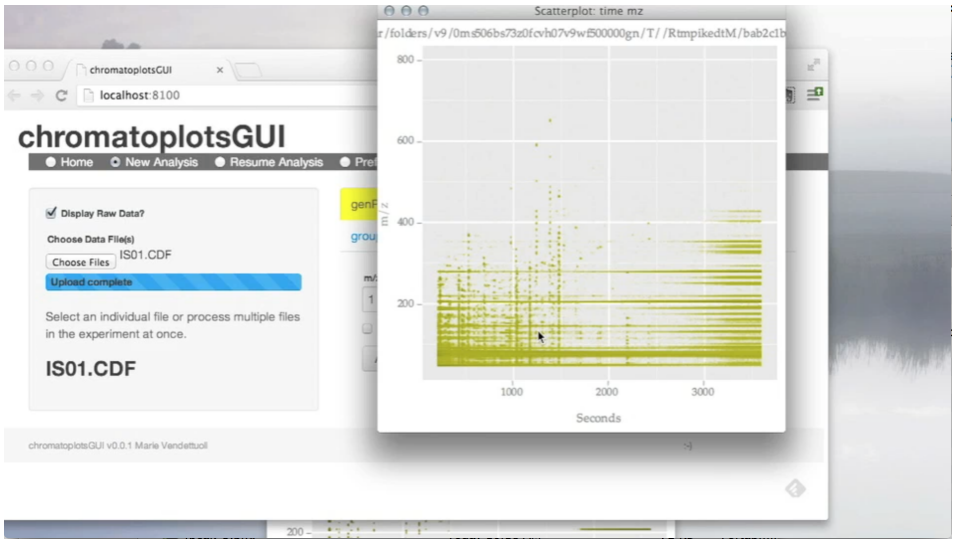


Figure 5.10: Using **chromatoplotsGUI** to display `genProfile` results

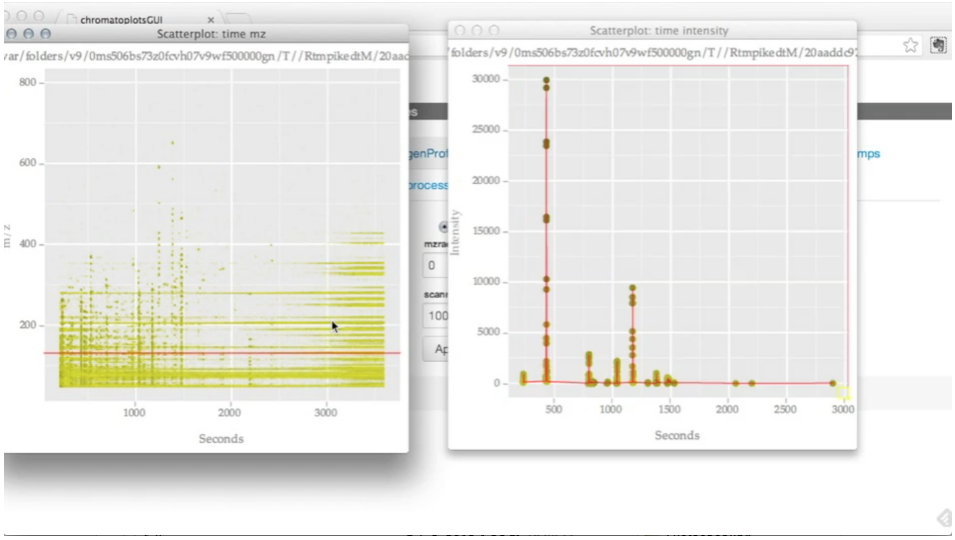


Figure 5.11: After removeBaseline step has been performed

CHAPTER 6. CONCLUSIONS

6.1 Significant contributions of dissertation

The framework presented in this dissertation has origins in reducing the amount of rework necessary for workflow development when implementing the technical solutions presented in each of three case studies. In the process of creating and implementing pipelines useful for each of these highly specialized audiences, it became apparent that existing models for development neglect basic concepts from fields of human computer interaction, interactive graphical software and reproducible research. Furthermore, this neglect transcends the size of the affected community or the perceived quality of the scientists involved. At one level, the case studies with their diversity of of target audiences are demonstrations of the broader impact of this research. Additionally:

6.1.1 ggparallel

The research in this paper addresses the issue of missing user testing for the popular visualization method parsets. Although the parsets approach has been written up both in academic and mainstream media, testing for the effectiveness of parsets's ability to communicate the underlying data has not been previous presented.

Technical implementation of *ggparallel* is in distribution on `CRAN` and at `https://github.com/heike/ggparallel`. User testing was conducted in survey form under IRB approval and may be found at: `http://vrac.us2.qualtrics.com/SE/?SID=SV_a35sfAgwuhewc6x`. The research paper associated with this case study was accepted by *IEEE Transactions on Visualization and Computer Graphics* June 2013, is reproduced in Chapter 2 and includes the following components:

1. Survey of existing options for displaying parallel coordinates of categorical data.

2. Implementation of parallel set and hammock plot algorithms in package *ggparallel*.

3. Propose a new display, common angle plots and implement in package *ggparallel*.

4. Perform usability testing to evaluate performance of this new display against existing tools.

5. Propose a quantitative measurement of the line-width illusion, a lie factor effect which explains reduced user performance when reading information encoded in hammock plots and parallel set charts.

6.1.2 chromatoplotsGUI

The motivation for development of chromatoplotsGUI is to address the need for analysis of metabolomic data in a reproducible and extensible environment that involves interactive visualizations. Commercial tools (e.g. Chemstation) may embed interactive graphics, but limit the user to analytic algorithms specified by the developer. Open source tools (e.g. xcms) may allow for extensibility, but offer limited options (none of which are interactive) for displaying data graphically.

Underlying infrastructure for the interactive graphics of *chromatoplotsGUI* is the new `R` package *cranvas* [Xie et al. (2012)]. At this time, there are only trivial examples of using *cranvas* for exploratory data analysis. *chromatoplotsGUI* may act as a teaching example. Package *chromatoplots* is newly developed and the development of the GUI supports its validation.

Chapter 4 discusses testing to validate the chromatoplots engine and preliminary coding for visualizations and interface may be found at: `https://github.com/mariev/chromatoplotsgui`. Release of *chromatoplotsGUI* will follow outlet selected for *chromatoplots*, which has yet to be determined. While *chromatoplotsGUI* can be used independently of chromatoplots, with xcms as the analysis engine, greater flexibility regarding the curve fits matching across spectra is enabled. The project consists of the following elements:

1. Validate *chromatoplots* using data generated from samples with known content.

2. Create interactive visualizations in R for all stages of processing in *chromatoplots*, individually

3. Create a graphical user interface that incorporates visualizations created into a single framework that illustrates concepts underlying metabolomics analysis

6.1.3 Veterinary Biologics

While the tools developed are for daily use by statisticians at CVB, expectations of formatting data submissions affect the workflow of over 300 firms in industry. For smaller firms and operators without access to a mature data management team, the published guidance is an educational resource.

The paper summarizing this research may be found in Chapter 3. The first `R` package, *PF* is in distribution on `CRAN`. Additional packages *dataFormats* and *CVBreports* may be found at https://github.com/mariev. Documentation associated with the data formatting standards are available at: http://www.aphis.usda.gov/animal_health/vet_biologics/vb_data_formats.shtml.

1. Develop data format standards. Document and communicate to internal and external stakeholders via hand-on training, email consultations, publicly accessible web pages and help pages. Use of these data formating standards have reduced the amount of rejected data submissions from approximately 50% at project start to current rates of <5%.

2. Create R packages to transform raw data (in specified data formats) into visualizations and summaries useful for preliminary data analysis. Package *PF* is currently available on CRAN. Packages *dataFormats* and *CVBreports* are distributed via github.

3. Create LaTeXtemplates for the processing of routine reports to meet business objectives.

4. Perform a qualitative study to evaluate impact of these tools in the CVB Statistics group. Turnaround time for initial processing of data submissions is now on the order of 48 hours versus historical turnaround times of up to 4 months.

6.2 Ongoing development

One major opportunity for future development is the inclusion of browser-based interactive graphical output. Interactivity provides an opportunity to further reduce cognitive load when viewing data displays. With transitions between visualization of shorter duration, the demand on short-term memory is reduced. Likewise, if interactivity is tied to exploratory activities (e.g. the ability reordering by different statistical summaries or at the user's unique prioritization scheme) the dependency on an individual's working memory is lowered, freeing resources to focus on higher-order analysis.

6.2.1 Interactivity: ggparallel and Veterinary Biologics

For *ggparallel*, this would mean converting existing functions which use static displays of type *ggplot2* to Javascript or relying on Qt engines. Of the two the former is an attractive solution because there is no need to install additional software. What is also attractive is the possibility to leverage R packages that are currently in development, such as *shiny* or *r2d3* (Wickham, 2012). The ability to deploy visualizations without the need for individual users to have a local installation of R is of great value in diverse working groups such as CVB Statistics where infrastructure limitations mean that it is highly problemmatic for rapid deployment to non R users.

6.2.2 Usability testing

For *chromatoplotsGUI* the next logical step is to perform usability testing with the validation data set. Of particular interest is optimization of the GUI to reduce wait times associated with underlying processing algorithms. While the time needed to perform calculations associated with each step of the analysis process is ultimately the domain of the underlying engine *xcms* or *chromatoplots*, providing immediate feedback to users and using non-optimal engineering to provide a superior experience (Krieger, 2011).

APPENDIX A. OTHER PUBLICATIONS

A.1 Clustering microarray data to determine normalization method

a chapter in *Software Tools and Algorithms for Biological Systems*

This paper [Vendettuoli et al. (2011)] originated as a collaborative term project in `STAT430` (Instructor: H. Hofmann) with classmate Erin Doyle (BCB) during Fall 2008. In Fall 2009 I revised the paper, verified reproducibility and submitted the final version to editors.

Abstract

Most of the scientific journals require published microarray experiments to meet Minimum Information About a Microarray Experiment (MIAME) standards. This ensures that other researchers have the necessary information to interpret the results or reproduce them. Required MIAME information includes raw experimental data, processed data, and data processing procedures. However, the normalization method is often reported inaccurately or not at all. It may be that the scaling factor is not even known except to experienced users of the normalization software. We propose that using a seeded clustering algorithm, researchers can identify or verify previously unknown or doubtful normalization information. For that, we generate descriptive statistics (mean, variance, quantiles, and moments) for normalized expression data from gene chip experiments available in the ArrayExpress database and cluster chips based on these statistics. To verify that clustering grouped chips by normalization method, we normalize raw data for chips chosen from experiments in ArrayExpress using multiple methods. We then generate the same descriptive statistics for the normalized data and cluster the chips using these statistics. We use this dataset of known pedigree as seeding data to identify normalization methods used in unknown or doubtful situations.

Background

ArrayExpress is a publicly accessible database of transcriptomics microarray data maintained by the European Bioinformatics Institute (EBI) and consists now of over 250,000 microarray assays. One major objective of EBI is to make data freely available to the scientific community, under the guidelines of the Microarray and Gene Expression Data (MGED) Society, which has defined standards for MIAME (Minimum Information About a Microarray Experiment). Three of the criteria required by MIAME are (a) the distribution of raw data (b) the distribution of normalized data and (c) annotation regarding the normalization protocol. Because normalization methods are an essential component of data collection and processing, it is imperative that annotation is accurate. Additionally, the process of normalization is time-consuming and computationally intensive, motivating researchers to identify efficient approaches beyond repetition to ensure veracity of publicly available data.

In this paper, we use cluster analysis to determine methodology used for normalization of microarray data. We evaluate this technique against normalized data retrieved from ArrayExpress and against raw data normalized by the authors using the most common normalization methods of Affymetrix Analysis Suite v5.0 (Mas5) and Robust Multichip Average (RMA). As this question first arose to researchers focused on Arabidopsis thaliana, the data used for this analysis will be drawn from experiments using this model organism.

Normalization

Converting raw data from microarray experiments to gene expression values requires a considerable amount of data processing, including normalization. Normalization ensures that all of the chips are comparable to each other and removes systemic effects of experimentation [Gautier et al. (2004)]. Data may be log transformed during normalization so that absolute changes in value represent a fold change in expression. Popular algorithms RMA and Mas5 perform normalization as part of a three-step process flanked by background subtraction and summarization, respectively [Gautier et al. (2004)]. RMA assumes a common mean background, using perfect match data. After subtracting this value, intensities are adjusted for identical

distributions: data undergoes logarithmic transformation followed by quantile scaling [Irizarry et al. (2003)]. In contrast, Mas5 performs background subtraction using a localized value generated from the lowest 2% of data in a zone, with weighted averages based on distance as a smoothing factor. Summaries are calculated by first subtracting ideal mismatch intensities, log transforming the result, then scaling by a trimmed mean. This expression value is further modified by plotting intensities for each chips probes against a baseline (also scaled using trimmed mean), fitting a regression line to the middle 98%, and modifying the experimental probesets intensities so that this regression line becomes the identity x = y. Due to the complexity of this process, there is slight variation of results between Mas5 calculations performed on different platforms and even between using different compilers [Lim et al. (2007)].

Clustering

Clustering is used to divide data objects into meaningful groups. Ideally, objects that are most closely related to each other are placed in the same cluster, and objects that are dissimilar are placed in different clusters.

Agglomerative hierarchical clustering begins with each object in an individual cluster. At each step of the algorithm, the closest or most similar clusters are joined [Tan et al. (2005)]. Results are displayed as a dendrogram that shows order in which objects were joined. Similarity is determined by a distance measure, with Euclidean most commonly used. As objects are joined into clusters, this distance matrix must be updated, requiring a method for defining the distance between clusters, in addition to individuals. For average-linkage clustering, all pair-wise distances between two clusters are calculated and the average is used as the distance [Pevsner (2009)]. Wards method aims to maximize homogeneity within each group by minimizing variation [Hådle and Simar (2003)].

Data processing methods

All processing was initially performed using R version 2.9.2 [R Core Team (2012)]. RMA and Mas5 normalization was performed using affy package version 1.2 [Gautier et al. (2004)]. Packages cluster [Maechler et al. (2005)] and pvclust [Suzuki and Shimodaira (2011)] were used

to calculate and annotate dendrograms. Visualization makes use of package ggplot2 [Wickham (2010a)]. Data, both raw and pre-normalized, were obtained from ArrayExpress, and contained both Affymetrix chips and samples with no array design information available.

Pre-normalized dataset

Normalized gene expression data for 50 chips was downloaded from ArrayExpress. Each chip was taken from a different experiment. Most chips listed Affymetrix MA software for the normalization, but not version or scaling factor. For each chip, the following descriptive statistics were calculated from normalized expression values: mean, min, max, variance, skewness, kurtosis, and 10th through 90th quantiles.

Author-normalized dataset

Raw data from 11 randomly selected experiments were downloaded from ArrayExpress. Five chips from each experiment were selected to be normalized. The chips were normalized in groups of 5 with all chips in a group coming from the same experiment. Additionally, the number of chips selected for each normalization process varied to mitigate impact of experimental design symmetry on results. We chose to group and normalize the chips in this way to reflect a batch process that researchers are likely to encounter.

Mas5 55 chips (taken from 11 experiments) were normalized twice, once using scale factor 500 and again using scale factor 250. In addition, 25 of these chips were randomly selected and normalized with a scale factor 100.

RMA normalization starts with quantile, cyclic loess, or contrast normalization followed by summarization in log base two. We chose quantile normalization because it is the most rapid of the three options, minimizing the need for excessive computing. 55 chips were normalized using this method in groups of 5 as described previously, with 25 of these chips randomly selected from those already included in the Mas5 group. Descriptive statistics were computed for each author-normalized chip as described above.

Clustering

Chips were clustered based on the computed descriptive statistics using iterative calculations of distances according to Wards method. Elements of the tree are organized with the tightest cluster (smallest distance) to the left. It is important to note that with different datasets, it is expected that new relationships (due to additional chips) will change the hierarchical ordering. By applying a p-value filter to the tree, we were able to designate clusters that the data strongly supported. P-values were obtained by multiscale bootstrap resampling with n=1000.

Observations

Author-normalized Dataset (Training)

Descriptive statistics We used a parallel coordinate plot to examine the frequency of specific values for each descriptive statistic. For some descriptors, such as variance, a casual observer cannot identify which normalization method corresponds to a specific value. Other descriptors mean, median show obvious grouping by normalization method; however it is difficult to distinguish

boundaries. After examining all descriptive statistics (data not shown) we determined that the distribution of 90th quantile gave the greatest separation between normalization methods (Figure A.1a). For all descriptive statistics RMA normalization shows the least variation in values. The increasing distance between normalization methods for higher level quantiles reflects the range to which each approach scales the chips. A greater scaling factor magnifies variation in observed values.

The shape and confidence interval for descriptive statistics is supported by the central limit theorem: normalized data has an approximately normal distribution for the mean. For quantile values (median, 90th quantile) distribution is normal for the middle quantiles, with increasing right-side skew as one increases past the 75th quantile [Sitter and Wu (2001); David and Nagaraja (2003)].

Cluster analysis The dendrogram shown in figure A.1b lists a subset of the results of clustering all author normalized chips. Visual inspection reveals a perfect grouping of microarrays into separate clusters along normalization method. A more robust analysis, by designating a p-value cutoff, generates a list of clusters. The benefit of visual analysis is that relationships are immediately apparent for all chips clustered. However, when examining datasets from more than 50 chips the display may become too crowded to distinguish individual labels. Screening clusters by p-values ensures that all selected clusters meet confidence criteria for calculated distances. As number of chips involved in analysis increases, so does confidence level for the overall tree (from p=0.70 for 20 chips to p=0.95 for 40 chips).

Pre-normalized Dataset (Testing)

Descriptive statistics Plots of descriptive statistics for pre-normalized data shows that most of these values fall in the same

range as the statistics for the author-normalized data, with minimal outliers (Figure A.1a). Additional values are expected, as it is possible that data from the ArrayExpress database was normalized by methods not covered above. Documentation from ArrayExpress indicates that the four outliers were subject to ANOVA normalization.

In order to determine the normalization method for each grouping, we seeded pre-normalized data with values generated for use in the author-normalized section. After excluding the outliers, pre-normalized data aligns well with author-normalized values, especially when looking at the 90th quantile. 3.2.2 Cluster analysis While inspecting descriptive statistic plots allows us to qualitatively define relationships between pre-normalized data and author-normalized methods (Figure A.1b); performing cluster analysis as described above provides a quantitative way to measure the strength of the predicted groupings. The tree shown in Fig 4 indicates that pre-normalized chips one and eight have author processed, RMA-normalized chips as nearest neighbors. ArrayExpress documentation confirms these chips as being RMA normalized. Remaining pre-normalized chips cluster with Mas5 author-normalized values with various scaling factors. These clusterings also agreed with documentation in ArrayExpress. Additionally, it was observed that clustering is by normalization method irrespective of source data.

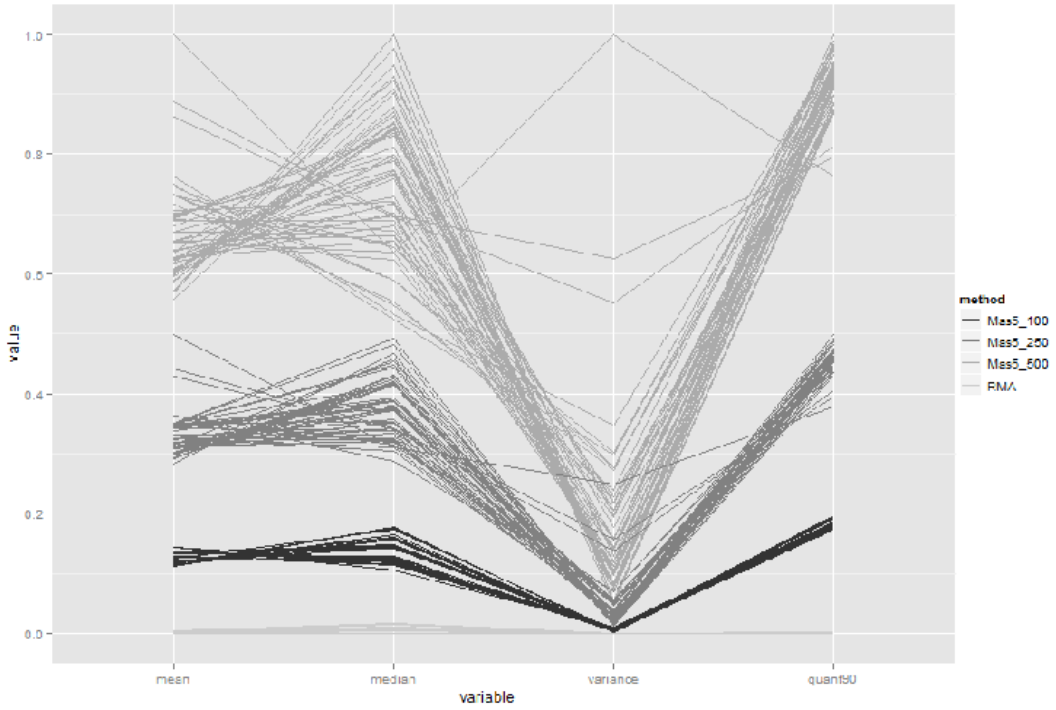
Mas5 scaling factors Although ArrayExpress allows researchers to specify that chips were normalized using Mas5, it does not provide a field to report the scaling factor used. Our results show that Mas5 normalized chips with different scaling factors tend to cluster by scaling factor. Additionally, researchers may need to perform analysis across sets of chips normalized using different scaling factors. Therefore, it is useful to be able to convert Mas5 normalized data from one scaling factor to another.

Plotting values from the same dataset normalized with different Mas5 scaling factors show a strong linear relationship (not shown). Therefore, Mas5 normalized data can be transformed from one scaling factor to another much more rapidly than the 5-10 min needed to re-normalize each .Cel file in R. Transformation is performed simply by multiplying the ratios of each scaling factor. To convert from scaling factor 250 to 500, we multiply by $(500/250) = 2$.

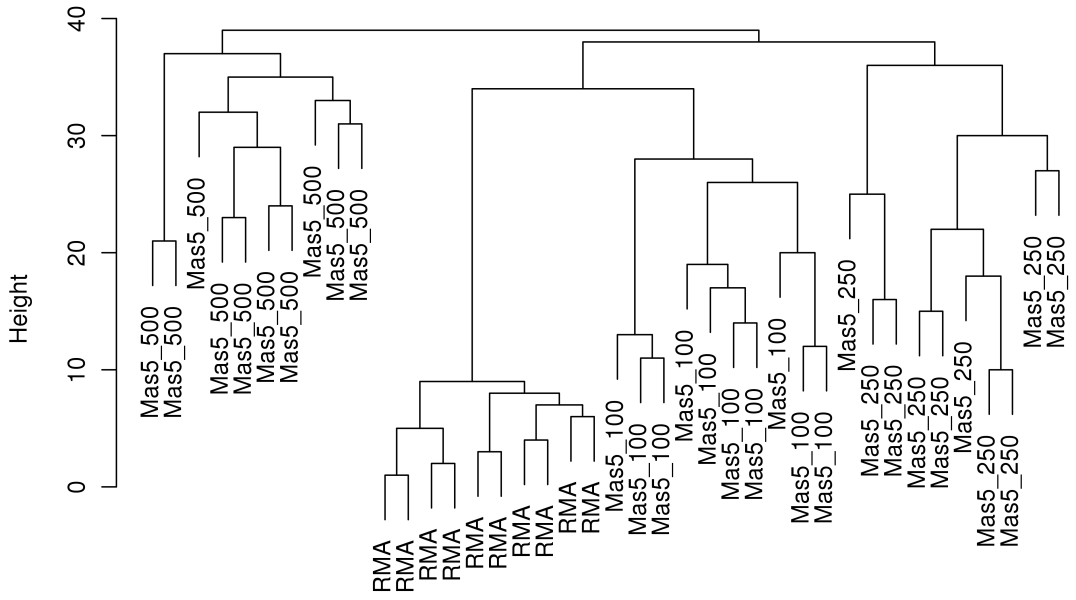
Conclusion

We demonstrated that different normalization processes are distinguishable by specific descriptive statistics at both visual and quantitative levels. Incorporation of data with known normalization alongside data of unknown normalization (seeding) is a straightforward and robust technique for a researcher to identify unknown normalization method. Transformation between Mas5 normalizations scaled to different values requires one to simply multiply by a constant value.

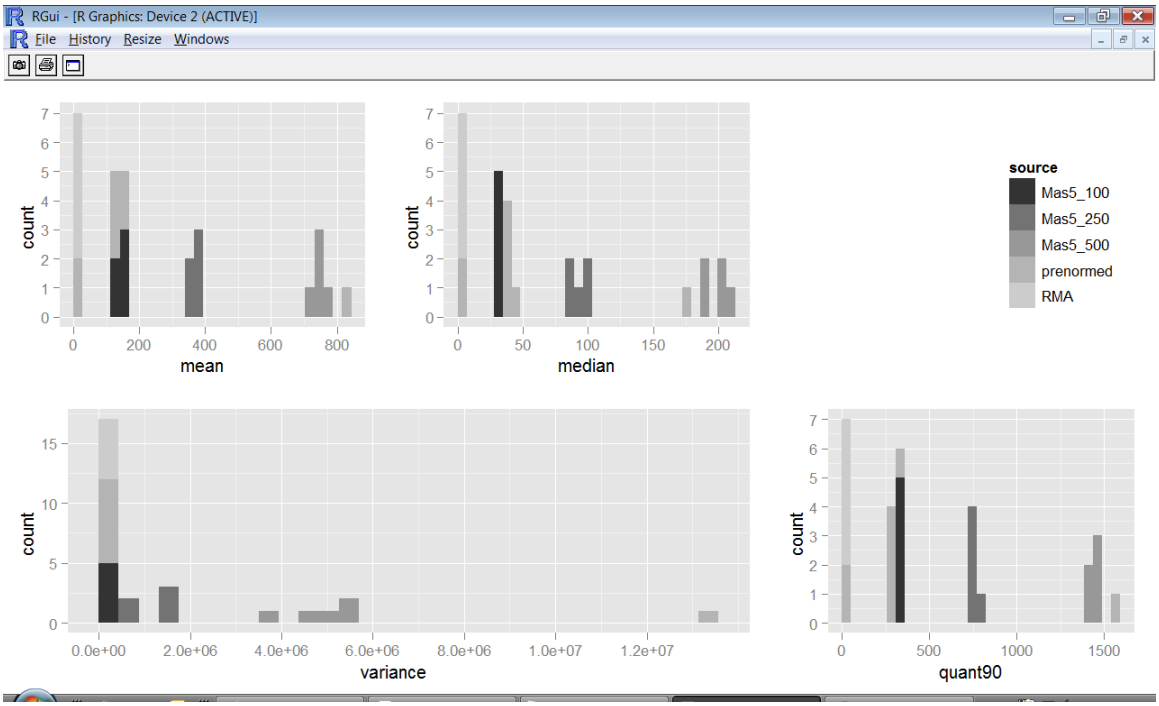
These actions (visualization, seeding, and transformation) together define a workflow for researchers to accurately and confidently identify specific normalization methods used to generate a set microarray data, without spending time replicating the normalization process themselves. This not only facilitates easier reproducing of previously published results, but it will also mediate novel discoveries as researchers will be able to utilize previously unusable data sets in ArrayExpress and other databases.



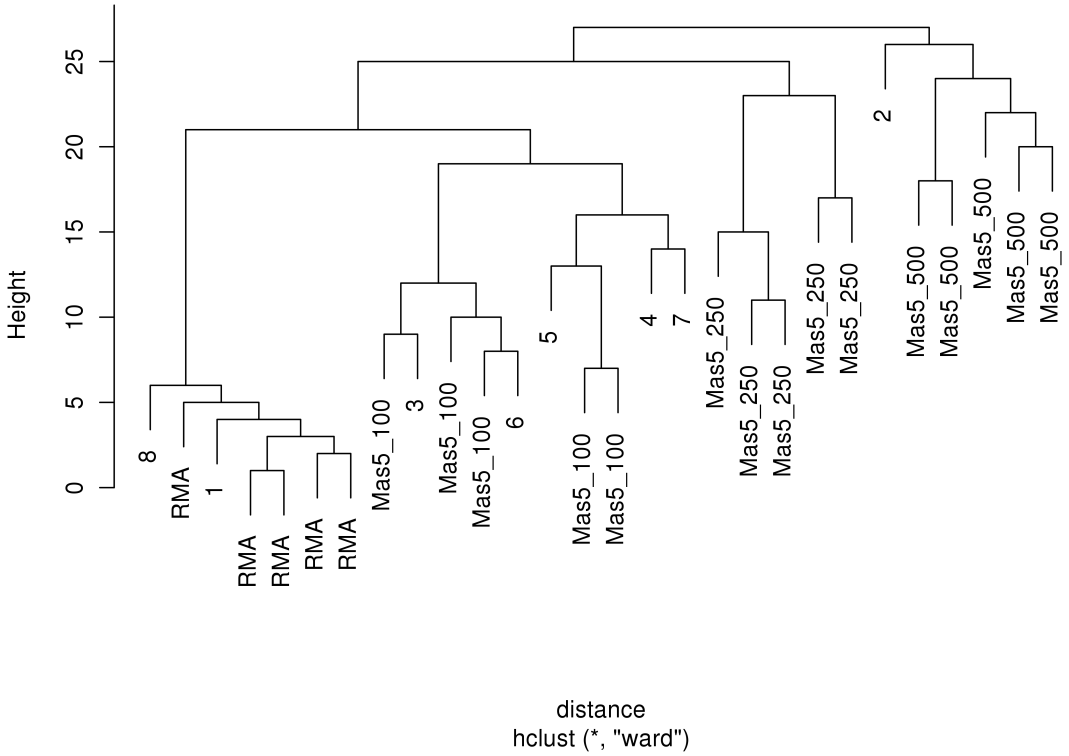
(a) Distribution of mean, median, variance and 90th quantile values obtained from author-normalized data.



(b) Dendrogram showing a subset of all author-normalized chips. All distances have p-value of 0.95. Splits are plotted at equally based heights.



(a) Descriptive statistics from subset of pre-normalized data seeded with that of author-normalized data.



(b) Subset of seeded data. Pre-normalized values are numbered 1 thru 8, author-normalized values are labeled by normalization method and scaling factor, if applicable. All distances have p-value of 0.95. Splits are plotted at equally based heights.

BIBLIOGRAPHY

Ahmad, I., Suits, F., Hoekman, B., Swertz, M., Byelas, H., Dijkstra, M., Hooft, R., Katsubo, D., van Breukelen, B., Bischoff, R., and Horvatovich, P. (2011). A high-throughput processing services for retention time alignment of complex proteomics and metabolomics lc-ms data. *Bioinformatics*, 27(8):1176–1178.

Andreev, V., Rejtar, T., Chen, H., Moskovets, E., Ivanov, A., and Karger, B. (2003). A universal denoising and peak picking algorithm for lc-ms based on matched filtration in the chromatographic time domain. *Anal. Chem*, 75(22):6314–26.

Apache (2012). Subversion.

Baggerly, K. and Coombes, K. (2009). Deriving chemosensitivity from cell lines: Forensic bioinformatics and reproducible research in high-throughput biology. *The Annals of Applied Statistics*, 3(4):1309–1334.

Bagozzi, R. (2007). *Journal of the Association for Information Systems*, 8(4):244 –254.

Baran, R., Kochi, H., Saito, N., Suematsu, M., Soga, T., Nishioka, T., Robert, M., and Tomita, M. (2006). Mathdamp: a package for differential analysis of metabolite profiles. *BMC Bioinformatics*, 7(1):530.

Bates, D., Maechler, M., and Bolker, B. (2012). *lme4: Linear mixed-effects models using S4 classes*. R package version 0.999999-0.

Bellew, M., Coram, M., Fitzgibbon, M., Igra, M., Randolph, T., Wang, P., May, D., Eng, J., Fang, R., and Lin, C. (2006). A suite of algorithms for the comprehensive analysis of complex protein mixtures using high-resolution lc-ms. *Bioinformatics*, 22(15):1902.

Bendix, F., Kosara, R., and Hauser, H. (2005). Parallel Sets: Visual analysis of categorical data. In *IEEE Symposium on Information Visualization*.

Benton, H. P., Want, E. J., and Ebbels, T. M. (2010). Correction of mass calibration gaps in liquid chromatography-mass spectrometry metabolomics data. *Bioinformatics*, pages 26 – 2488.

Blastland, M. (March 11 2009). Go figure: How to understand risk in 13 clicks.

Bostock, M., Ogievetsky, V., and Heer, J. (2011). D3: Data-driven documents. *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*.

Buckheit, J., Donoho, D., et al. (1995). Wavelab and reproducible research. *LECTURE NOTES IN STATISTICS-NEW YORK-SPRINGER VERLAG-*, pages 55–55.

Buja, A., Hurley, C., and MCDONALD, J. (1987). A data viewer for multivariate data. In *Colorado State Univ, Computer Science and Statistics. Proceedings of the 18 th Symposium on the Interface p 171-174(SEE N 89-13901 05-60)*.

Buja, A., McDonald, J., Michalak, J., and Stuetzle, W. (1991). Interactive data visualization using focusing and linking. In *Visualization, 1991. Visualization'91, Proceedings., IEEE Conference on*, pages 156–163. IEEE.

Carroll, J. (2009). *Human–Computer Interaction*. Wiley Online Library.

Carter, S. and Bostock, M. (Oct 15 2012). Over the decades, how states have shifted.

Cheng, X. (2011). Cranvastime: Interactive longitudinal and temporal data plots.

Christensen, J., Mortensen, J., Andersen, O., and Hansen, A. (2005). Chromatographic preprocessing of gc-ms data for analysis of complex chemical mixtures. *J. of Chromatography A*, 1062(1):113–123.

Cleveland, W. (2001). Data science: an action plan for expanding the technical areas of the field of statistics. *International Statistical Review*, 69(1):21–26.

Cleveland, W. S. and McGill, R. (1984). Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American Statistical Association*, 79(387):pp. 531–554.

Cook, D. and Swayne, D. (2007). *Interactive and Dynamic Graphics for Data Analysis: with R and GGobi*. Springer.

Dahl, D. B. (2013). *xtable: Export tables to LaTeX or HTML*. R package version 1.7-1.

Danielsson, R., Bylund, D., and Markides, K. (2002). Matched filtering with background supression for improved quality of base peak chromatograms and mass spectra in liquid chromatography-mass spectrometry. *Analytica Chimica Acta*, 454(2):167–184.

Davenport, T. H. and Patil, D. J. (2012). Data scientist: The sexist job of the 21st century. *Harvard Business Review*, 90(10):70–76.

David, H. and Nagaraja, H. (2003). *Order Statistics*. Wiley.

Davies, J. (2012). Parallel sets.

Davis, F. (1989). Perceived usefulness, perceived ease of use and user acceptance of information technology. *MIS Quarterly*, 3(3):319–340.

Dawson, R. J. (1995). The 'unusual episode' data revisited. *Journal of Statistics Education*, 3.

Day, R. H. and Stecher, E. J. (1991). Sine of an illusion. *Perception*, 20:49–55.

Denison, D. R. and Mishra, A. K. (1995). Toward a theory of organizational culture and effectiveness. *Organizational Science*, 6(2):204–223.

Do Yup Lee, B. P. B. and Northen, T. R. (2010). Mass spectrometrybased metabolomics, analysis of metabolite-protein interactions, and imaging. *Biotechniques*, 49(2):557–564.

Du, P., Kibbe, W., and Lin, S. (2006). Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching. *Bioinformatics*, 22(17):2059–2065.

Eilers, P. (2004). Parametric time warping. *Anal. Chem*, 76(2):404–411.

Fernando, R. and Pharr, M. (2004). *GPU gems: programming techniques, tips, and tricks for real-time graphics*, volume 697. Addison-Wesley.

Fiehn, O., Wohlgemuth, G., Scholz, M., Kind, T., Lee, D., Lu, Y., Moon, S., and Nikolau, B. (2008). Quality contro for plant metabolomics: reporting msi-compliant studies. *The Plant Journal*, 53(4):691–704.

Foley, J. and Van Dam, A. (1982). Fundamentals of interactive computer graphics. *Addison-Wesley Systems Programming Series, Reading, Mass.: Addison-Wesley, 1982*, 1.

Fox, J. (2003). Effect displays in r for generalised linear models. *Journal of Statistical Software*, 8(15):1–27.

Fox, J. (2009). Aspects of the social organization and trajectory of the R project. *The R Journal*, 1(2):5–13.

Fox, J. and Hong, J. (2009). Effect displays in r for multinomial and proportional-odds logit models: Extensions to the effects package. *Journal of Statistical Software*, 32(1):1–24.

Friendly, M. (1992). Visualizing categorical data: Data, stories and pictures. In *SAS User Group Conference*.

Fry, B. J. (2004). *Computational Information Design*. PhD thesis, Massachusetts Institute of Technology.

Gautier, L., Cope, L., Bolstad, B., and Irizarry, R. (2004). Affy–analysis of affymetrix genechip data at the probelevel. *Bioinformatics*, 20:215–307.

Gentleman, R. and Temple Lang, D. (2004). Statistical analyses and reproducible research.

Georgakopoulos, D., Hornick, M., and Sheth, A. (1995). An overview of workflow management: From process modeling to workflow automation infrastructure. *Distributed and Parallel Databases*, 3:119–153. 10.1007/BF01277643.

Git (2012).

Grunbacher, H. (1998). Teaching computer architecture/organisation using simulators. In *Frontiers in Education Conference, 1998. FIE'98. 28th Annual*, volume 3, pages 1107–1112. IEEE.

Harding, C. and Souleyrette, R. (2010). Investigating the use of 3d graphics, haptics (touch), and sound for highway location planning. *Computer-Aided Civil and Infrastructure Engineering*, 25(1):20–38.

Harris, P. A., Taylor, R., Thielke, R., Payne, J., Gonzalez, N., and Conde, J. G. (2009). Research electronic data capture (redcap)a metadata-driven methodology and workflow process for providing translational research informatics support. *Journal of Biomedical Informatics*, 42(2):377 – 381.

Harrower, M. A. and Brewer, C. A. (2003). ColorBrewer.org: An Online Tool for Selecting Color Schemes for Maps. *The Cartographic Journal*, 40(1):27–37.

Hartigan, J. and Kleiner, B. (1982). Mosaics for contingency tables. In *Proceedings Symposium on the Interface*.

Hastings, C., Norton, S., and Roy, S. (2002). New algorithms for processing and peak detection in liquid chromatography/mass spectrometry data. *Rapid Communications in Mass Spectrometry*, 16(5):462–467.

Heer, J. and Bostock, M. (2010). Crowdsourcing graphical perception: Using mechanical turk to assess visualization design. In *CHI 2010: Visualization*.

Hettne, K., Wolstencroft, K., Belhajjame, K., Goble, C., Mina, E., Dharuri, H., Verdes-Montenegro, L., Garrido, J., de Roure, D., and Roos, M. (2012). Best practices for workflow design: how to prevent workflow decay. In Paschke, A., Burger, A., Romano, P., Marshall, M. S., and Splendiani, A., editors, *Proceedings of the 5th International Workshop on Semantic Web Applications and Tools for Life Sciences (SWAT4LS2012)*.

Hoffmann, N. and Stoye, J. (2009). Chroma: signal-based retention time alignment for chromatography mass spectrometry data. *Bioinformatics*, 25(16):2080–2081.

Hofmann, H. (2000). Exploring categorical data: Interactive mosaic plots. *Metrika*.

Hofmann, H. and Vendettuoli, M. (2012). *ggparallel: Variations of Parallel Coordinate Plots for Categorical Data*. R package version 0.1.1.

Hornik, K. (2012a). Are there too many R packages. *Austrian Journal of Statistics*, 41(1):59–66.

Hornik, K. (2012b). The `R` FAQ.

Hornik, K. and Leisch, F. (2002). Vienna and R: Love, marriage and the future. pages 61–70.

Hothorn, T., Bretz, F., and Westfall, P. (2008). Simultaneous inference in general parametric models. *Biometrical Journal*, 50(3):346–363.

Hådle, W. and Simar, L. (2003). *Applied Multivariate Statistical Analysis*. Springer.

Hurley, C. (1987). *The Data Viewer: a program for graphical data analysis*. PhD thesis, University of Washington.

Irizarry, L., Hobbs, B., Collin, F., and Speed, T. (2003). Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, 4:240–264.

Jaffe, J., Mani, D., Leptos, K., Church, G., Gillette, M., and Carr, S. (2006). Pepper a platform for experimental proteomic pattern recognition. *Molecular and Cellular Proteomics*, 5(10):1927.

Johnson, K., Wright, B., Jarman, K., and Synovec, R. (2003). High-speed peak matching algorithm for retention time alignment of gas chromatographic data for chemometric analysis. *J. of Chromatography A*, 996(1):141–155.

Jr, A. M. and Jorgenson, J. (1993). Median filtering for removal of low-frequency background drift. *Anal. Chem*, 65(2):188–191.

Katajamaa, M. and Oresic, M. (2007). Data processing for mass spectrometry-based metabolomics. *J. of Chromatography A*, 1158:318–328.

Kent, W., Sugnet, C., Furey, T., Roskin, K., Pringle, T., Zahler, A., and Haussler, D. (2002). The human genome browser at UCSC. *Genome Research*, 12(6):996–1006.

Kind, T. and Fiehn, O. (2010). *Advances in structure elucidation of small molecules using mass spectrometry*, volume 2 of *Bioanalytical Reviews*. Springer.

Koehring, A., Foo, J., Miyano, G., Lobe, T., and Winer, E. (2008). A framework for interactive visualization of digital medical images. *Journal of Laparoendoscopic & Advanced Surgical Techniques*, 18(5):697–706.

Kohlbacher, O., Reinert, K., Gropl, C., Lange, E., Pfeifer, N., Schulz-Trieglaff, O., and Sturm, M. (2007). Topp the openms proteomics pipeline. *Bioinformatics*, 23(2).

Kong, N., Heer, J., and Agrawala, M. (2010). Perceptual guidelines for creating rectangular treemaps. *Transactions on Visualization and Computer Graphics*.

Kosara, R. (2008). Redesigning parallel sets. Visweek 2008 workshop.

Kosara, R. (2012). Eagereyes.

Kosara, R., Bendix, F., and Hauser, H. (2006). Parallel sets: Interactive exploration and visual analysis of categorical data. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):558–568.

Koutroumpas, M. and Higgins, C. (2008). In *11th AGILB International Conference on Geographic Information Science*.

Krieger, M. (2011). Secrets to lightening fast mobile design. Warm Gun.

Kroening, D. and Paul, W. (2001). Automated pipeline design. In *Design Automation Conference, 2001. Proceedings*, pages 810–815. IEEE.

Lan, K. and Jorgenson, J. (2001). A hybrid of exponential and gaussian functions as a simple model of asymmetric chromatographic peaks. *J. of Chromatography A*, 915(1-2).

Lawrence, M. (2006). *Interactive graphics, graphical user interfaces and software interfaces for the analysis of biological experimental data andnetworks*. PhD thesis, Iowa State University.

Lawrence, M., Yin, T., Hofmann, H., yeon Choi, S., and Cook, D. (2012a). *chromatoplots: Preprocessing of GC-MS metabolomics data with a GUI and interactive plots.* R package version 0.0.8.

Lawrence, M., Yin, T., Hofmann, H., yeon Choi, S., and Cook, D. (2012b). *chromatoplots: Preprocessing of GC-MS metabolomics data with a GUI and interactive plots.* R package version 0.0.9.

Leisch, F. (2002). Sweave: Dynamic generation of statistical reports using literate data analysis. In Härdle, W. and Rönz, B., editors, *Compstat 2002 — Proceedings in Computational Statistics*, pages 575–580. Physica Verlag, Heidelberg. ISBN 3-7908-1517-9.

Lencioni, P. (2002). *The Five Dysfunctions of a Team*. Jossey-Bass.

Li, X., Gentleman, R., Lu, X., Shi, Q., Iglehart, J., Harris, L., and Miron, A. (2005). *Bioinformatics and Computational Biology Soulutions Using R and Bioconductor*, chapter SELDI-TOF Mass Spectrometry Protein Data, pages 91–109. Springer.

Lim, W., Wang, K., Lefebvre, C., and Califano, A. (2007). Comparative analysis of microarray normalization procedures: effects on reverse engineering gene networks. 13:282–288.

Listgarten, J., Neal, R., Rowels, S., and Emili, A. (2005). Multiple alignment of continuous time series. *Advances in Neural Information Processing Systems*, 17:817–824.

Liu, W., Schmidt, B., Voss, G., and Müller-Wittig, W. (2008). Accelerating molecular dynamics simulations using graphics processing units with cuda. *Computer Physics Communications*, 179(9):634–641.

Lommen, A. (2009). Metalign: Interface-driven, versatile metabolomics tool for hyphenated full-scan mass spectrometry data preprocessing. *Anal. Chem*, 81(8):3079–3086.

Luse, A., Scheibe, K. P., and Townsend, A. M. (2008). A component-based framework for visualization of intrusion detection events. *Inf. Sec. J.: A Global Perspective*, 17(2):95–107.

MacFarlane, J. (2006). Pandoc, a universal document converter.

MacFarlane, J. (2012a). Daring fireball: Markdown.

MacFarlane, J. (2012b). Pandoc: a universal document converter.

MacKenzie, I. (1992). Fitts' law as a research and design tool in human-computer interaction. *Human-computer interaction*, 7(1):91–139.

Maechler, M., Rousseeuw, P., Struyf, A., Hubert, M., and Hornik, K. (2005). *cluster: Cluster Analysis Basics and Extensions*. R package version 1.14.1 — For new features, see the 'Changelog' file (in the package source).

Morris, J., Coombes, K., Koomen, J., Baggerly, K., and Kobayashi, R. (2005). Feature extraction and quantification for mas spectrometry in biomedical applications using the mean spectrum. *Bioinformatics*, 21(9):1764.

Neuwirth, E. (2011). *RColorBrewer: ColorBrewer palettes*. R package version 1.0-5.

Nielsen, N., Carstensen, J., and Smedsgaard, J. (1998). Aligning of single and multiple wavelength chromatographic profiles for chemometric data analysis using correlation optimized warping. *J. of Chromatography A*, 805(1-2):17–35.

Nordvik, T. and Harding, C. (2008). Interactive geovisualization and geometric modelling of 3d data-a case study from the åknes rockslide site, norway. *Headway in Spatial Data Handling*, pages 367–384.

Patterson, D. A. and Hennessy, J. L. (2011). *Computer Organization and Design*. Morgan Kaufmann, 4 edition.

Pevsner, J. (2009). *Bioinformatics and Functional Genomics*. Wiley-Blackwell.

Playfair, W. (1786). *Commercial and Political Atlas*. London.

Playfair, W., Wainer, H., and Spence, I. (2005). *Playfair's Commercial and Political Atlas and Statistical Breviary*. Cambridge University Press.

Pluskal, T., Castillo, S., Villar-Briones, A., and Oresic, M. (2010). Mzmine 2: Modular framework for processing visualizing and analyzing mass spectrometry-based molucular profile data. *BMC Bioinformatics*, 11:395.

R Core Team (2012). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

Robbins, N. (2005). *Creating More Effective Graphs*. Wiley.

Robbins, S. P. and Judge, T. A. (2010). *Organizational Behavior*. Prentice Hall, 14 edition.

Rogers, S., Scheltema, R., Barrett, M., and Breitling, R. (2011). *Handbook of Statistical Systems Biology*, chapter Bayesian Approaches for Mass Spectrometry-Based Metabolomics. Wiley.

Rosson, M. and Carroll, J. (2001). *Usability engineering: scenario-based development of human-computer interaction*. Morgan Kaufmann.

Ruckstuhl, A., Jacobson, M., Field, R., and Dodd, J. (2001). Baseline subtraction using robust local regression estimation. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 68(2):179–193.

Salfner, F. and Pablant, N. (2009).

Sauve, A. and Speed, T. (2004). Normalization, baseline correction and alignment of high throughput mass spectrometry data. In *Proceedings of the Genomic Signal Processing and Statistics*.

Schonlau, M. (2003). Visualizing categorical data arising in the health sciences using hammock plots. In *Proceedings of the Section on Statistical Graphics*. RAND Corporation, American Statistical Association.

Shah, R., Chandrasekaran, A., and Linderman, K. (2008). In pursuit of implementation patterns: The context of lean and six sigma. *International Journal of Production Research*, 46(23):6679–6699.

Sharpe, S. W., Johnson, T. J., Sams, R. L., Chu, P. M., Rhoderic, G. C., and Johnson, P. A. (2004). Gas-phase databases for quantitative infared spectroscopy. *Applied Spectroscopy*, 58(12).

Siev, D. (2012). *PF: Functions related to prevented fraction.* R package version 9.4.

Sitter, R. and Wu, C. (2001). A note on woodruff confidence intervals for quantiles. *Statistics & Probability Letters*, 52:353–358.

Smith, C.A., Want, E.J., O'Maille, G., Abagyan,R., Siuzdak, and G. (2006). Xcms: Processing mass spectrometry data for metabolite profiling using nonlinear peak alignment, matching and identification. *Analytical Chemistry*, 78:779–787.

Smith, C., Want, E., O'Maille, G., Abagyan, R., and Siuzdak, G. (2006). Xcms: processing mass spectrometry data for metabolite profiling using nonlinear peak alignment, matching and identification. *Anal. Chem*, 78(3):779–787.

Stein, S. (1999). An integrated method for spectrum extraction and compound identification from gas chromatography/mass spectrometry data. *J. Am. Soc. Mass Spectrom*, 10(8):770–781.

Sutherland, P., Rossini, A., Lumley, T., Lewin-Koh, N., Dickerson, J., Cox, Z., and Cook, D. (2000). Orca: A visualization toolkit for high-dimensional data. *Journal of Computational and Graphical Statistics*, 9(3):509–529.

Suzuki, R. and Shimodaira, H. (2011). *pvclust: Hierarchical Clustering with P-Values via Multiscale Bootstrap Resampling.* R package version 1.2-2.

Swayne, D., Cook, D., and Buja, A. (1991). Xgobi: interactive dynamic graphics in the x window system with a link to s.

Swayne, D., Cook, D., and Buja, A. (1998). Xgobi: Interactive dynamic data visualization in the x window system. *Journal of Computational and Graphical Statistics*, 7(1):113–130.

Swayne, D., Lang, D., Buja, A., and Cook, D. (2003). Ggobi: Evolving from xgobi into an extensible framework for interactive data visualization. *Computational Statistics & Data Analysis*, 43(4):423–444.

Symanzik, J., Majure, J., and Cook, D. (1996). Dynamic graphics in a gis: A bidirectional link between arcview 2.0 and xgobi.

Tan, P., Steinbach, M., and Kumar, V. (2005). Introduction to data mining.

Tautenhahn, R., Boettcher, C., and Neumann, S. (2008). Highly sensitive feature detection for high resolution lc/ms. *BMC Bioinformatics*, 9:504.

Tenllado, C., Setoain, J., Prieto, M., Piñuel, L., and Tirado, F. (2008). Parallel implementation of the 2d discrete wavelet transform on graphics processing units: Filter bank versus lifting. *Parallel and Distributed Systems, IEEE Transactions on*, 19(3):299–310.

Theus, M., Hofmann, H., Siegl, B., and Unwin, A. (1997). *New Techniques and Technologies for Statistics II*, chapter MANET: Extensions to interactive statistical graphics for missing values. IOS Press Amsterdam.

Tufte, E. (1991). *The Visual Display of Quantitative Information.* Graphics Press, USA, 2 edition.

Tukey, J. W. (1980). We need both exploratory and confirmatory. *The American Statistician*, 34(1).

Qt Project (2011). Download Qt.

van der Aalst, W., ter Hofstede, A., Kiepuszewski, B., and Barros, A. (2003). Workflow patterns. *Distributed and Parallel Databases*, 14:5–51. 10.1023/A:1022883727209.

van der Aalst, W. and van Hee, K. M. (2004). *Workflow Management: Models, Methods, and Systems.* MIT Press, 2 edition.

Venatesh, V., Morris, M., Davis, G., and Davis, F. (2003). User acceptance of information technology: Toward a unified view. *MIS Quarterly*, 27(3):425 –478.

Vendettuoli, M. (2012). *dataFormats: Import and Export of CVB Data Formats*. R package version 2.24.

Vendettuoli, M. (2013). *CVBreports: Create Reports for CVB Statistics*. R package version 0.0-6.22.

Vendettuoli, M., Doyle, E., and Hofmann, H. (2011). *Clustering microarray data to determine normalization method*. Advances in Experimental Medicine and Biology. Springer.

Vendettuoli, M. and Hofmann, H. (2012). Interactive hammock plots for visualizing categorical data. In *UseR!2012*.

Venkatesh, V. (2000). Determinants of perceived ease of use: Integrating control, intrinsic motivation, and emotion into the technology acceptance model. *Information Systems Research*, 11(4):342 –365.

Venkatesh, V. and Bala, H. (2008). Technology acceptance model 3 and a research agenda on interventions. *Decision Sciences*, 39(2):273 – 315.

Venkatesh, V. and Davis, F. (2000). A theoretical extension of the techology acceptance model: Four longitudinal field studies. *Management Science*, 46(2):186–204.

Wainer, H. (2000). *Visual Revelations*. Psychology Press.

Walkenbach, J. (2010a). *Excel 2010 Bible*. Wiley Publishing.

Walkenbach, J. (2010b). *Excel 2010 Power Programming with VBA*. Wiley Publishing, 1 edition.

Wang, P., Tange, H., Fitzgibbon, M., Mcintosh, M., coram, M., Zhang, H., Yi, E., and Aebersold, R. (2007). A statistical method for chromatographic alignment of lc-ms data. *Biostatistics*, 8(2):357.

Wang, W., Zhou, H., Lin, H., Roy, S., Shaler, T., Hill, L., Norton, S., Kumar, P., Anderle, M., and Becker, C. (2003). Quantification of proteins and metabolites by mass spectrometry without isotopic labeling or spiked standards. *Anal. Chem*, 75(18):4818–4826.

Ward, J. H., J. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58:236–244.

Wickens, C. D., Lee, J. D., Liu, Y., and Becker, S. E. G. (2004). *An introduction to human factors engineering*. Pearson Prentice Hall, 2 edition.

Wickham, H. (2007). Reshaping data with the reshape package. *Journal of Statistical Software*, 21(12):1–20.

Wickham, H. (2009). *ggplot2: elegant graphics for data analysis*. Springer New York.

Wickham, H. (2010a). A layered grammar of graphics. *Journal of Computational and Graphical Statistics*, 19(1):3–28.

Wickham, H. (2010b). stringr: modern, consistent string processing. *The R Journal*, 2(2):38–40.

Wickham, H. (2012). *r2d3: Tools for making d3 visualisations from R*. R package version 0.0.1.

Wickham, H. and Hofmann, H. (2011). Product plots. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2223–2230.

Wickham, H., Lawrence, M., Cook, D., Buja, A., Hofmann, H., and Swayne, D. (2009). The plumbing of interactive graphics. *Computational Statistics*, 24(2):207–215.

Wilkinson, L. (2005). *The grammar of graphics*. Springer.

Womack, J. P., Jones, D. T., and Roos, D. (2007). *The Machine that changed the world*. Free Press.

Xie, Y. (2012). *knitr: A general-purpose package for dynamic report generation in R*. R package version 0.8.

Xie, Y. (2013). *knitr: A general-purpose package for dynamic report generation in R*. R package version 1.2.

Xie, Y., Hofmann, H., Cook, D., Cheng, X., Schloerke, B., Vendettuoli, M., Yin, T., Wickham, H., and Lawrence, M. (2012). *cranvas: Interactive statistical graphics based on Qt*. R package version 0.8.2.

Yin, T. (2011). Visnab: An interactive toolkit for visualizing and exploring genomic data.

Yu, W., Li, X., Liu, J., Wu, B., Williams, K., and Zhou, H. (2006). Multiple peak alignment in sequential data analysis: A scale-space based approach. In *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, volume 3(3), pages 208–219.

Zack, M., McKeen, J., and Singh, S. (2009). Knowledge management and organizational performance: an exploratory analysis. *Journal of Knowledge Management*, 13(6):392–409.