Development of Bioinformatics Tools for Biomedical High-Throughput Analyses

List, Markus

Publication date: 2015

Citation for published version (APA):

Terms of use
This work is brought to you by the University of Southern Denmark through the SDU Research Portal. Unless otherwise specified it has been shared according to the terms for self-archiving. If no other license is stated, these terms apply:

• You may download this work for personal use only.
• You may not further distribute the material or use it for any profit-making activity or commercial gain.
• You may freely distribute the URL identifying this open access version.

If you believe that this document breaches copyright please contact us providing details and we will investigate your claim. Please direct all enquiries to puresupport@bib.sdu.dk.
Development of Bioinformatics Tools for Biomedical High-Throughput Analyses

A thesis submitted in fulfilment of the requirements for the degree of Ph.D.
in the

Group of Molecular Oncology
Lundbeckfonden Center of Excellence in Nanomedicine
Institute of Clinical Research
Faculty of Health Sciences

June 2015
“Learning Is A Treasure That Will Follow Its Owner Everywhere”

Chinese Proverb
Abstract

Ph.D.

Development of Bioinformatics Tools for Biomedical High-Throughput Analyses

by Markus List

Despite of improvements in treatment and diagnosis, breast cancer is still the major cause of death in women world-wide. Highly resilient breast cancer stem cells have been implicated in treatment resistance, metastasis and tumor relapse, demanding the development of new targeted therapies. The combination of large-scale functional genomics techniques and high-throughput screening (HTS) is a powerful approach to systematically discover drug targets and novel agents. In one application of HTS, the effect of silencing each individual gene in the genome is studied via RNA interference. A differential screen in triplicates involves approximately 120,000 samples, which can be further utilized in a secondary readout on the protein level. Here, each sample is typically spotted more than 20 times in a dilution series on so-called reverse-phase protein arrays (RPPAs). Interrogating these for only 10 cancer markers thus leads to 24 million sample spots. Pivotal to the success of drug target discovery strategies are efficient bioinformatics methods for sample management, data processing and analysis, as well as systems biology methods to generate new testable hypotheses. To address these challenges, suitable bioinformatics web applications were developed in the frame of this thesis.

OpenLabFramework (OLF) is a laboratory information management system covering all aspects of experimental documentation and sample management in high-throughput functional genomics. OLF is linked to Sample Management and Visual Analysis of HTS (SAVANAH), which provides sample management of molecular libraries and thousands of microtiter plates. Raw HTS data is normalized, accounting for known sources of variation, and evaluated in RNAi and compound screen evaluation (RNAice). Furthermore, RNAice provides functional enrichment analyses and addresses particular challenges for microRNA and small compound screens, where target genes need to be determined prior to analysis. Finally, Microarray R-based Analysis of Complex Lysate Experiments (MIRACLE) allows for management and analysis of RPPA data, addressing particular challenges in this technology to obtain protein concentration estimates from dilution series.

In conclusion, the tools presented here enable researchers to handle millions of samples across different high-throughput platforms. Sample management is streamlined and
highly efficient and in addition, functions for processing, analysis, and visualization are available. Each of the developed tools can be used alone or as integral part of a high-throughput drug discovery platform, which has the potential to significantly accelerate the development of new treatment options in breast cancer.
Dansk Resume

Ph.d.

Etablering af Bioinformatiske Værktøjer til Biomedicinsk
High-Throughput Data Analyse

fra Markus List

Til trods for fremskridt inden for behandling og diagnosticering, er brystkræft stadig hovedårsagen til dødsfald blandt kvinder verden over. Særligt fleksible brystkræftstamceller er blevet impliceret i behandlingsresistens, metastasedannelse, samt tilbagevendende tumorer, hvilket kræver udvikling af nye målrettede behandlingsstrategier. Kombinationen af omfattende funktionelle genomteknikker og high-throughput screening (HTS) er en effektiv tilgang til systematisk at identificere gener til målrettet behandling. I én anvendelse af HTS, undersøges effekten når alle individuelle gener i genomet hæmmes enkeltvis via RNA interferens. Et differentielt screen i triplikater indeholder ca. 120.000 prøver, hvilke yderligere kan anvendes til sekundær udlæsning på proteinniveau.

Her spottes hver prøve typisk mere end 20 gange i en fortyndingsserie af såkaldte reverse-phase protein arrays (RPPAs). Dette resulterer i 24 millioner prøver, så fremt hver enkelt skal undersøges for bare 10 cancermarkører. Afgørende for succesen af denne strategi til identifikation af behandlingsmuligheder er effektive metoder inden for bioinformatik til at varetage prøverne, databehandling og analyse, samt system biologi til at generere nye hypoteser. For at gribe disse udfordringer an, udvikledes egnede bioinformatiske webapplikationer i omfanget af denne disputats.


Afslutningsvis sætter disse værktøjer forskere i stand til håndterer millioner af prøver over forskellige high-throughput platforme. Varetagelsen af prøver er strømlinet og
særdeles effektiv og i tilgift er funktioner til behandling, analysering, og visualisering tilgængelige. Hvert af de udviklede værktøjer kan anvendes alene eller som integrered del af en platform til high-throughput identifikation af behandlingsmidler, hvilket potentielt set signifikant kan accelerere udviklingen af nye behandlingsmuligheder i brystkræft.
Acknowledgements

When I and my wife Susanna moved to Denmark in August 2011, we embarked on three different journeys to learn new things. We wanted to experience life abroad, start a family, and I began to work on a Ph.D. degree. Naturally, the past four years were exciting in more than one way. We found many new friends and enjoyed living in Denmark, where we embraced the concept of “hygge”. Two wonderful daughters, Zoe and Lara, made our lives richer. I have learned a lot in these past four years and now that two of our journeys come to an end, there are many people I owe gratitude.

First of all, I would like to thank my supervisors Qihua Tan, Jan Mollenhauer, and Jan Baumbach, who each motivated me in their own way to tackle the challenges that lay ahead of me. I am deeply grateful for their support and for the faith they had in my abilities.

I would like to thank all of my colleagues for creating a work environment that made it fun to do a PhD even in stressful times. Thanks to all present and past members of the Mollenhauer group, Ines, Sidsel, Monica, Cinzia, Daniel, Steffen, Helle, Pernille, Carolin, Martina, Matthias, Jakub, Aleksandra, and Angela. A special thanks goes to Marlene, for being patient with bug fixes and for being willing to learn R to meet me half-way. Moreover, thanks to all the members of the Baumbach group, Nicolas, Richa, Anne-Christin, Lucas, Diogo, Christian, Eudes, Anders, Martin, and Michael. Furthermore, I am grateful for useful advice and practical help I received from Fabio Vandin, Richard Röttger, Mads Thomassen, and Torben Kruse.

There are countless friends we made during our time here and I would like to name a few who have made our time in Odense as lovely as it was. Thanks to Torben, Linda, Peter, Jule, Mathias, Petra, Heidi, Paul, Dennis, Thorsten, Olena, Manuela, Diyako, Bartek, Agnieska, Abbas, Elena, Asif, and Justyna. Living far away from your home makes you realize which are the most important friends in your life. Thanks Paula-Jo, Niki and Günter for always being there for me. A special thanks to Günter for his invaluable comments on this thesis from an outside perspective.

Most importantly, I would like to thank my parents, who have selflessly supported me throughout my entire life without expecting anything in return. To them and to my parents in law, I am grateful that they spent countless hours on the road, just to be with us and our children. Last, but definitely not least, I want to say thank you, Susanna. This adventure would not have been possible without you. I thrive through your love and support.
2 Classification of Breast Cancer Subtypes by Combining Gene Expression and DNA Methylation Data

2.1 State of the Art

2.2 Material & Methods

2.2.1 Data

2.2.2 Classification

2.2.2.1 Random Forest and Bootstrapping

2.2.2.2 Feature Importance

2.2.3 Model Evaluation

2.3 Results

2.3.1 Classification Performance of the Different Models

2.3.2 Analysis of Misclassification by the Control Model

2.3.3 Most Important Features

2.3.4 Overlap of Feature Lists

2.4 Discussion & Conclusion

3 OpenLabFramework: A Laboratory Information Management System with Focus On Functional Genomics

3.1 Requirements

3.1.1 Implementation

3.1.2 Data Handling

3.1.3 Flexibility in Deployment

3.1.4 User Acceptance and Excess Value

3.1.5 Security

3.2 State of the Art

3.3 Results

3.3.1 Modularity and Extendibility

3.3.2 Sample Management

3.3.3 Sample Storage

3.3.4 File Uploads

3.3.5 Reporting

3.3.6 Deployment

3.3.7 Barcode and Label Support

3.3.8 Web Application Interface and Documentation

3.3.9 Typical Work-Flow

3.4 Discussion & Conclusion

4 OpenLabNotes: An Electronic Laboratory Notebook Extension for OpenLabFramework
5 SAVANAH: Sample Management and Visual Analysis of High-Throughput Screens 51
5.1 Requirements ................................................. 53
5.2 State of the Art .............................................. 54
5.3 Results ....................................................... 54
5.3.1 Plate Layouts .............................................. 55
5.3.2 Plates and Readout Import ............................... 55
5.3.3 Projects and Experiments ............................... 56
5.3.4 Screening Libraries ..................................... 56
5.3.4.1 Import Library ........................................ 56
5.3.4.2 Browse Library ........................................ 57
5.3.4.3 Library Dilutions .................................... 57
5.3.5 Create Experiment from Library ..................... 58
5.3.6 Data Analysis ............................................. 59
5.4 Discussion & Conclusion ................................. 59

6 RNAice: High-Throughput RNAi and Compound Screen Evaluation for R 61
6.1 Requirements ................................................ 62
6.1.1 Data Input ................................................ 62
6.1.2 Quality Control .......................................... 62
6.1.3 Normalization and Hit Discovery ...................... 63
6.1.4 Downstream Analysis .................................... 63
6.1.4.1 MicroRNAs ........................................... 63
6.1.4.2 Small Compounds .................................... 64
6.1.4.3 Functional Enrichment Analysis .................. 64
6.1.5 User Interface ............................................ 64
6.2 State of the Art ............................................. 64
6.3 Methods ...................................................... 65
6.3.1 Normalization of Raw Signal ......................... 65
6.3.1.1 Control Based Normalization ..................... 66
6.3.1.2 Plate Based Normalization ....................... 66
6.3.2 Quality Control .......................................... 68
6.3.2.1 Strictly Standardized Mean Difference ........ 68
6.3.3 Hit Detection ............................................. 69
6.3.3.1 SSMD for Hit Detection ......................... 69
6.3.3.2 Bayesian Hit Detection ......................... 70
6.3.4 MicroRNA target genes ............................... 71
6.3.4.1 High-Confidence miRNA Target Genes ........ 72
6.3.5 Analysis of Candidate Genes ......................... 73
6.3.5.1 Gene Set Overrepresentation Analysis .......... 73
6.3.5.2 Gene Set Enrichment Analysis .................. 73
6.3.5.3 Network Enrichment Analysis ................... 74
6.4 Results ....................................................... 75
6.4.1 User Interface ............................................ 75
6.4.1.1 Input .................................................. 75
6.4.1.2 Data .................................................. 77
6.4.1.3 Quality Control ........................................ 77
6.4.1.4 Hit Discovery .......................................... 80
6.4.2 Application Cases ........................................ 83
6.4.2.1 A Genome-Wide RNA Interference Screen Identifies Cas-
pase 4 as a Factor Required for Tumor Necrosis Factor
Alpha Signaling ............................................ 83
6.4.2.2 Genome-Wide Functional Genomics Analysis for Genes
Regulating Sensitivity to Vorinostat ........................ 86
6.5 Discussion & Conclusion ................................... 87

7 MIRACLE: Microarray R-based Analysis of Complex Lysate Experi-
ments .............................................................. 91
7.1 Requirements .................................................. 92
7.1.1 Sample Management ....................................... 92
7.1.2 Analysis ..................................................... 93
7.2 State of the Art ............................................... 93
7.3 Methods ......................................................... 94
7.3.1 Raw Signal Normalization ................................. 94
7.3.2 Quantification ............................................... 94
7.3.3 Loading Control Normalization ........................... 95
7.4 Results .......................................................... 95
7.4.1 Sample Management ....................................... 96
7.4.1.1 Plate and Slide Layouts ................................. 96
7.4.1.2 Virtual Spotting ........................................ 98
7.4.1.3 Projects and Experiments ............................... 98
7.4.2 Data Processing .............................................. 98
7.4.2.1 Slides .................................................. 98
7.4.2.2 Supported File Formats ................................. 99
7.4.2.3 Processing Raw Data ................................... 99
7.4.2.4 Plates and Readouts ................................... 99
7.4.3 Data Analysis ................................................ 100
7.4.3.1 Processing of Raw Signal: .............................. 100
7.4.3.2 Protein Concentration Estimates and Sample Grouping . 100
7.4.3.3 Comparison across Slides and Readouts ............... 102
7.4.3.4 Correlation with Primary Readout .................... 102
7.4.3.5 Significance Testing .................................... 102
7.4.3.6 Import and Export .................................... 102
7.5 Discussion & Conclusion .................................... 103

8 Implementation .................................................. 107
8.1 Grails Applications for Sample Management ................. 107
8.1.1 Grails Web Application Framework ....................... 107
8.1.1.1 Domain Classes ....................................... 108
8.1.1.2 Controller and Service Classes ....................... 108
8.1.1.3 Groovy Server Pages ................................ 109
8.1.2 Application Security ....................................... 109
8.1.2.1 Data Export to R via Web Service .................... 109


8.1.3 Implementation Details ........................................ 110
8.1.3.1 OpenLabFramework ........................................ 110
8.1.3.2 OpenLabNotes ............................................. 112
8.1.3.3 Shared Functionality between SAVANAH and MIRACLE 113
8.1.3.4 Plate and Slide Layouts .................................... 115
8.1.3.5 Plates and Slides .......................................... 115
8.1.3.6 Layout Editor .............................................. 115
8.1.3.7 Grails OpenSeaDragon Plug-in .......................... 115
8.2 R shiny Applications for Data Analysis and Visualization .... 116
8.2.1 R shiny ......................................................... 116
8.2.2 Plotting ........................................................ 117
8.2.3 Code Structure ................................................ 117

9 Discussion and Conclusion ........................................... 119
9.1 Classification of Breast Cancer Subtypes by Combining Gene Expression and DNA Methylation Data ...................... 120
9.2 OpenLabFramework: A Laboratory Information Management System with Focus On Functional Genomics .................. 121
9.3 OpenLabNotes: An Electronic Laboratory Notebook Extension for Open-LabFramework ........................................ 123
9.4 SAVANAH: Sample Management and Visual Analysis of High-Throughput Screens .............................................. 124
9.5 RNAice: High-Throughput RNAi and Compound Screen Evaluation for R 126
9.6 MIRACLE: Microarray R-based Analysis of Complex Lysate Experiments ......................................................... 128
9.7 An Integrated Functional Genomics Platform for Drug Discovery .... 130

10 Outlook ................................................................. 133

Bibliography ............................................................... 137
List of Figures ............................................................ 154
List of Tables ............................................................. 157
Abbreviations ............................................................. 159
A Supplemental Tables .................................................. 163
B Supplemental Figures .................................................. 171
C Availability .................................................................. 181
C.1 Version Control System and Public Code Repository ........ 181
C.2 Open Source License .............................................. 181
C.3 Classification of Breast Cancer Subtypes By Combining Gene Expression and DNA Methylation Data ...................... 182
C.4 OpenLabFramework .............................................. 182
C.5 OpenLabNotes ....................................................... 182
C.6 HTSbackend ........................................................... 183
Contents

C.7 SAVANAH ................................................................. 183
C.8 RNAice ................................................................. 183
C.9 MIRACLE ............................................................... 184

D Supplemental Methods ................................................. 185
D.1 Details on Bayesian Hit Detection ................................. 185
D.2 Authentication to Grails Applications in R ....................... 188
Dedicated to the memory of my grandfather...
Chapter 1

Introduction

1.1 Motivation

Despite of recent advances in prognosis and therapy, breast cancer remains the second most common cause of cancer-related death in women, accounting for 23% of total cancer cases and 14% of cancer deaths [1].

Traditional therapies, such as surgical intervention, chemotherapy, and radiation therapy often fail to capture all tumor cells, leading to tumor recurrence and metastasis, in which the cancer cells enter the blood stream to spread to other organs with fatal outcome [2]. So-called cancer stem cells (CSCs) are suspected to be the main reason for treatment resistance. In addition, different molecular breast cancer subtypes exist, which differ in prognosis and treatment requirements [3]. Classifying breast cancer patients into these subtypes enables precision medicine in which molecular properties of the disease are exploited with the aim of increasing specificity and efficacy of treatment [4].

To facilitate the development of precision medicine, a better understanding of cancer development is needed. Systematic high-throughput approaches are well suited to identify causal cancer genes, for example, from next-generation sequencing (NGS) or messenger RNA (mRNA) expression data. Such functional genomics approaches are feasible through advanced cell manipulation techniques. These allow creating large panels of isogenic cell lines that only differ in the molecular change under investigation. These approaches are particularly powerful in combination with high-throughput screening (HTS), in which a large number of experiments can be performed in parallel to study phenotypical changes and / or to identify genotype-selective vulnerabilities of cancer cells. Massive amounts of experimental data from these and other high-throughput approaches can then be analyzed individually or in combination to gain new insights into driving molecular mechanisms and potential new therapeutic strategies for cancer.
However, a common theme of high-throughput approaches in general is that they demand advanced bioinformatics methods for sample management and data analysis. Therefore, the goal of the project underlying this thesis was to deliver intuitive and efficient bioinformatics tools to cover several challenges reaching from functional genomics studies to HTS and array-based proteomics using reverse-phase protein arrays (RPPAs). Integrating these approaches into a drug discovery pipeline will, in addition, reduce the efforts necessary to keep track across several complex experiments. In this chapter, we illuminate some basics of the current knowledge of cancer development, as well as some key obstacles in the treatment of breast cancer, which is the focus of this work.

1.2 Cancer

1.2.1 Development

As they continue to replicate, cells inevitably suffer from damage to the DNA, which is typically detected and repaired. In cancer cells, however, genes involved in DNA repair are frequently mutated. This allows cancer cells to accumulate mutations and genetic variations such as insertions, deletions, gene duplication and chromosomal aberrations [5]. However, most of the mutations in a cancer cell are passenger mutations that do not have a significant effect on cell behavior. Cancer development is triggered by a set of so-called driver mutations responsible for a step-wise failure of the regulation of various cellular processes. Commonly, dysfunctional cells are programmed to die from apoptosis. Resisting this process is thus one of the most important hallmarks of cancer [6, 7]. Other hallmarks include the ability to sustain unlimited cell replication, to evade the immune system and to manipulate the local environment, for instance, by stimulating the growth of new blood vessels. While the tumor grows, some cancer cells may enter the bloodstream through a process, where epithelial to mesenchymal transition (EMT) could play a role. An EMT causes the cells to change to an invasive phenotype that is eventually suited for the conditions found in the vascular system. This process, together with the reverse process, mesenchymal to epithelial transition (MET), is considered as important for metastatic spread [8]. A better understanding of cancer progression is imperative to develop drugs that specifically target cancer vulnerabilities. However, cancer is a heterogeneous disease that ideally requires treatment optimization on a case by case basis. This is in essence the future promise of precision medicine (Chapter 1.3). In current clinical practice, different cancers are divided into subtypes defined through morphological and molecular characteristics of the malignant cells. The following chapter will elaborate on known subtypes in breast cancer and their links to the natural development of mammary tissue.
1.2.2 Breast Cancer Subtypes

Breast cancer is a heterogeneous disease, in which both treatment and prognosis depend largely on the subtype of the tumor. Due to its morphology, breast cancer can be divided into the basal and luminal subtype. The histological status of estrogen and progesterone receptor, as well as HER2 expression, are used as diagnostic markers to further differentiate luminal A, luminal B and the HER2 overexpressing subtype from the basal subtype, which is also known as triple-negative subtype [9]. A better understanding of the differences between these subtypes was achieved through the analysis of gene expression data, which lead to the distinction of five major subtypes, i.e. luminal A and B, basal, HER2 as well as the normal-like subtype [10, 11]. Subtyping based on gene expression was subsequently successfully commercialized leading to clinical tests such as Oncotype DX™ (Genomic Health Inc., Redwood City, CA, USA), MammaPrint™ (Agendia, Amsterdam, NL) or PAM50™ (Prosigna, Seattle, WA, USA). These tests rely on a set of 50-70 genes that were shown to have prognostic potential with regards to the clinically relevant subtypes. Larger sample numbers allowed later studies to define more fine-grained subtypes, which, however, remain poorly described such that the research focus remains on the established subtypes [12], with the notable exception of the claudin low subtype [13].

Progress in understanding the development of normal mammary tissue allowed linking the different molecular subtypes of breast cancer to various types of mammary cells (Figure 1.1) [14]. Here, the most aggressive subtypes, such as claudin-low and basal breast cancer, closely resemble stem cells and progenitor cells in their gene expression profile. In contrast, the less aggressive luminal types resemble the more differentiated epithelial and myoepithelial cells. Moreover, the frequency of the different subtypes matches the relative abundance of these cell types in breast tissue, making luminal A breast cancer the most frequent subtype. Evidence suggests that the breast cancer subtypes originate from mutations and genetic rearrangements occurring in stem or progenitor cells [15, 16]. In theory, this establishes a link between the origin of the breast cancer subtypes and the cancer stem cell model introduced in the following chapter. Although this link is controversial, it could be shown that the cancer stem cell content in fact correlates with the level of differentiation inherent to the different subtypes [17].

1.2.3 Cancer Stem Cells

Cancer stem cells (CSCs), also known as tumor initiating cells, have been implicated in treatment resistance and tumor relapse. CSCs represent a small sub-population of cancer cells that share properties of normal stem cells, such as asymmetric cell division,
Figure 1.1: Breast cancer can be divided into molecular subtypes that are linked to the different stages of cellular development of the mammary tissue. Moreover, the frequency and aggressiveness of the subtypes is reflected by the differentiation and relative abundance of these cell types. Adopted from [14].

which allows them to grow new tumors in which the majority of cells are again non-CSCs (Figure 1.2) [7, 18]. CSCs have been shown to express EMT markers [19], which makes them the primary suspects for the cause of the metastatic spread of the disease. The fact that EMT and MET have been observed in stem cells and progenitor cells in embryonic development indicates that this is a naturally occurring phenomenon that escaped regulation in cancer progression [20].

There is consensus that future therapies must target CSCs to fully eradicate the primary tumor and to prevent metastasis [3]. Several factors complicate such an endeavor. CSCs appear to exist in multiple forms and have even been shown to emerge through de-differentiation of bulk tumor cells [22]. This requires a set of highly specific drugs that act in combination to eradicate all types of tumor cells by exploiting their genetic vulnerabilities.
1.3 Precision Medicine

While some of the commonly de-regulated pathways in cancer are already known, it is not clear, without knowledge of the DNA sequence and the epigenetic patterns of the tumor, which particular mutations and epigenetic alterations caused a tumor to emerge and which of several redundant molecular mechanisms are affected. This has tremendous impact for selecting an effective treatment strategy, since the molecular properties of the tumor need to be understood on a patient-by-patient basis. Originally, this principle was therefore coined personalized medicine. However, since this can be misleading, the term precision medicine has been suggested as a replacement, highlighting that instead of offering a different drug for each patient, molecular properties of the patient’s tumor are exploited more efficiently than in the past [23].

With the rise of NGS technologies and the steady decrease in both time and costs necessary to sequence a human genome, obtaining the full sequence of a patient’s tumor will soon be a feasible option and thus enter clinical practice [24]. First of all, however, a better understanding of exploitable tumor properties, and of CSCs in particular, is needed. This is the goal of drug target screening and is intended to lead to the development of novel drugs that can then be used to eradicate all tumor cells, including CSCs, with high precision [3]. The key obstacle in this, is to make sure that non-cancerous cells remain mostly unharmed. To this end, the concept of synthetic lethality can be
utilized [25].

1.4 Synthetic Lethality

In synthetic lethality, two factors, such as two genes, are exploited in combination to achieve a lethal outcome. Let us assume a pair of genes provides the same essential function. Let us further assume that one of the two genes can be inactivated, e.g. through a drug. In this case the remaining active gene can still provide the essential functionality to the cell and the cells can continue to survive. If, however, both genes were inactivated simultaneously, no compensation mechanism would be left to provide the essential functionality and the cells would die.

Cancer is a disease in which cellular regulation mechanisms are disturbed, a fact necessary for cancer to develop. To maintain this status, cancer cells depend on mutations in molecular key pathways, which suggests that it is possible to find a scenario where one of two synthetic lethal factors is already inactivated through a mutation. Such a mutation may lead directly or indirectly to the hyperactivation of so-called oncogenes, i.e. genes that are implicated in cancer development and progression. This phenomenon is also referred to as oncogene addiction, due to the importance of these genes for cancer survival, and can be exploited for targeted anti-cancer therapy [26]. Synthetic lethal screening aims at finding a drug that can kill cancer (stem) cells specifically and to leave healthy (stem) cells intact (Figure 1.3). For a subset of breast cancer patients, this strategy proved already successful. Here, one of two DNA repair mechanisms is inactivated due to a mutation in the BRCA1/2 genes. Inhibiting the gene PARP through a drug removes the only alternative DNA repair mechanism, which causes cancer cells to die leading to tumor regression [27]. In the past, however, the discovery of such synthetic lethality scenarios proved difficult due to a lack of suitable techniques for cell manipulation at the genetic level [28]. Such techniques are presently available and allow for large scale functional genomics studies.

1.5 High-Throughput Functional Genomics

The aim of functional genomics is to study the function of genes at a substantial systematic scale. In this context, the term gene can refer to protein coding genes, as well as to microRNAs (miRNAs) or long non-coding RNAs (lncRNAs). In general, functional genomics can be divided into loss-of-function (LOF) and gain-of-function (GOF) studies [29]. In the following, the major methods for functional genomics studies are presented.
In the concept of synthetic lethality, two genes A and B provide an essential function to the cell in a redundant fashion. Inactivating one of the two genes will thus not affect healthy cells, but might prove lethal for cancer cells that do not possess an active copy of the second gene for compensation.

1.5.1 Loss-of-function Studies

LOF studies allow functional characterization by blocking either the expression or the function of a gene. In a gene knock-out experiment, a gene is rendered permanently dysfunctional. This can be achieved using transposons or retroviruses that target DNA sequences through homologous recombination to introduce a mutation. Alternatively, a gene knock-down, implying a transient inactivation of a gene, can be performed. Transient inactivation can be achieved by RNA interference (RNAi), while permanent inactivation or introduction of point mutations can be achieved via more sophisticated genome editing methods.

1.5.1.1 RNA interference (RNAi)

The RNAi pathway was first observed in 1990 in plants, where a trans-gene was suspected to have a gene silencing effect [30]. Nearly ten years later, Fire et al. could unravel the mechanism of RNAi and the role of double-stranded RNA in Caenorhabditis elegans, for which they received the Nobel prize in 2006 [31]. RNAi is a cellular process aimed at regulating the endogenous levels of messenger RNA (mRNA) in a cell and thus effectively regulates the amount of protein that can be translated. The RNAi pathway is mainly triggered by endogenous so-called miRNAs, which can down-regulate or silence up to several hundreds or even thousands of target genes by repressing the translation of the gene’s transcribed messenger RNA. This makes miRNAs the most important key players in cellular regulation next to transcription factors. Apart from miRNAs, the RNAi pathway can be triggered by small interfering RNAs (siRNAs), which naturally occur in the defense against viral DNA. In contrast to miRNAs, siRNAs typically have a specific target. It was soon discovered that this can be exploited to achieve nearly
complete gene knock-downs in LOF studies. The RNAi pathway is illustrated in Figure 1.4 and will be explained in more detail in the following two chapters.

1.5.1.2 MicroRNAs

Changes in miRNA expression have been implicated in breast cancer progression [32] and several miRNAs have been shown to be of major importance in breast cancer [33]. This suggests a potential not only as diagnostic and prognostic biomarkers, but also for the development of targeted therapies after successful drug target screening [34].

Mature miRNAs are 19 to 25 nucleotide long non-coding RNAs that function in the endogenous RNAi mediated repression of mRNAs. After transcription, miRNAs are generated via a series of processing steps. Primary miRNAs (pri-miRNAs) are first transcribed by RNA Polymerase II, leading to 70-100 nucleotide long hairpins known as precursor miRNA (pre-miRNA). Overhanging single stranded ends are subsequently cleaved by the endonuclease Drosha, before the precursors are transported from the nucleus into the cytoplasm by Exportin-5. In the cytoplasm, the loop structure is cleaved by the endonuclease Dicer, such that only a short-lived miRNA duplex remains. Only one of the two strands in this duplex, either the 3p or 5p variant, is usually active and is integrated as guide strand in the RNA induced silencing complex (RISC). The complementary strand, also known as passenger strand, is usually degraded. In the RISC complex, the guide strand interacts with the mRNA target site by base pairing. Perfect or almost perfect complementarity is only achieved in the so-called seed region at the 5' end of the miRNA, while a bulge seems to be required in the central region of the miRNA-mRNA duplex. Moreover, a good complementarity at the 3' end of the miRNA is advantageous to finally mediate the endonucleolytic cleavage of the mRNA through a member of the Argonaute protein family, which is part of the RISC complex. The miRNA-mediated post-transcriptional control is explained exhaustively in [35].

1.5.1.3 Small interfering RNAs

As described in the previous chapter, miRNAs achieve perfect complementarity only in the seed sequence. This allows them to target a large number of different genes. In contrast, siRNAs achieve perfect complementarity to their target mRNA over the entire length. Naturally, they occur as non-coding RNA duplexes or hairpins of the same length as miRNAs. They have been implicated in the cell’s defense against viruses, which often operate through double stranded RNAs to manipulate their host cells. After their initial discovery, it became soon clear that synthetic siRNAs can be utilized to specifically silence genes [36]. This triggered a massive research interest in siRNAs for influencing
cellular gene expression, as well as their potential use for targeted therapeutics. They proved particularly useful for HTS scenarios and have been successfully applied in many genome-wide LOF studies [37, 38]. However, caution is advised, since several studies indicate that siRNAs can suffer from off-target effects. This implies that a change in phenotype can not always be clearly attributed to the target gene that was intended to be inactivated [39]. Several strategies have been proposed to circumvent this issue, including pooling siRNAs with different sequences, control experiments to detect true and false positives, as well as several bioinformatics approaches. These strategies are reviewed in detail in [40].

**Figure 1.4:** The RNA interference (RNAi) pathway is utilized by miRNAs and other RNAi reagents. Single stranded ends of the pri-miRNA are cut by Drosha. The resulting pre-miRNA is exported out of the nucleus, where Dicer cleaves the loop away. The short-lived double stranded miRNA is separated and the active miRNA strand enters the RISC complex, where it leads to the recognition of a target mRNA sequence and to translational repression. In contrast to miRNAs, which recognize up to several hundreds of target genes through imperfect matching, small interfering RNA (siRNA) show perfect complementarity to the target sequence and lead to cleavage and immediate degradation of the mRNA. SiRNAs can start as small hairpin RNAs, long double-stranded RNA (dsRNA), synthetic siRNA (siRNA), or endoribonuclease-prepared siRNA (esiRNA). Adopted from [40, 41]

### 1.5.1.4 Genome Editing Methods

In the past few years, genome editing methods have become an attractive alternative, which can be used to modify or edit existing genes directly within a living cell. One
example of such a method are transcription activator-like effector nucleases (TALENs). These bind to genomic sequences flanking a target site and introduce double strand breaks. In consequence, a repair method known as nonhomologous end-joining is triggered, which, however, is error-prone and introduces small insertions or deletions in the target sequence. The resulting frame shifts are likely to cause a knock-out of the gene [42]. Alternatively, a donor template can be used to trigger a homology-directed repair pathway. The gene at this locus can thus be edited, allowing for a specific mutation to be studied. Moreover, this technique can in principle be used to repair dysfunctional genes by exchanging a mutated site with a non-mutated template, suggesting a huge potential for gene therapy. Another example of a similar genome editing method are zinc-finger nucleases (ZFNs), which consist of two domains, a zinc-finger domain that facilitates targeting to a specific DNA site as well as a nuclease domain that also introduces a double strand break triggering DNA repair [43]. A more recent genome editing method is the clustered regularly interspaced short palindromic repeats (CRISPR)/Cas9 system.

1.5.1.5 CRISPR/Cas9

CRISPR are a central part of a bacterial defense system in which invading DNA from viruses or plasmids is cut into small fragments and incorporated into a CRISPR locus flanked by a series of short repeats of around 20 base pairs in length. Transcripts of this locus are processed to generate CRISPR RNAs (crRNAs), which, together with a trans-activating crRNA (trRNA) serves as a guide for endonucleases. Three similar CRISPR mechanisms are known to date. In contrast to type I and III, however, type II depends only on a single effector protein, an endonuclease called Cas9. The crRNA facilitates targeted cleavage of the invading DNA through sequence complementarity, together with a short conserved sequence of 2 - 5 nucleotides length known as protospacer-associated motif (PAM), which follows immediately 3' of the crRNA complementary sequence. Jinek et al. were able to recruit this mechanism as a tool for functional genomics by combining crRNA and trRNA into a single synthetic single-guide RNA (sgRNA) [44]. The simplicity of the method has quickly made the CRISPR/Cas9 system a popular choice for LOF studies. Mutated versions of Cas9 allow for further applications, such as Cas9D10A, which cuts only one of the strands and can be used for homologous directed DNA repair via a donor template [45]. Another promising application allows in-vivo visualization using a version of Cas9 that has been fused with an enhanced green fluorescence protein [46]. In contrast to TALENs and ZFNs, where the DNA-protein interface needs to be adapted for each target gene, the CRISPR/Cas9 system requires only the design of a suitable sgRNA. This makes it easy to engineer and thus a powerful
alternative to RNAi technology for genome-wide LOF screening. Considering the issue of potential off-target effects, Mohr et al. suggest to complement RNAi screening with CRISPR/Cas9 to achieve a higher confidence in the observed effects [40].

1.5.2 Gain-of-function Studies

In contrast to LOF studies, GOF studies employ an over-expression system to study the effect of expressing a particular gene beyond its natural expression level or to express genes that do not yet exist in the target organism’s genome. The process of delivering DNA or RNA through target’s cell membrane is called transfection. Even naked DNA or RNA can be transfected through physical methods such as micro injection or electroporation. Chemical methods involve wrapping the negatively charged nucleic acids in a positively charged complex using, for instance, suitable lipids. Even though the details are not known, it is believed that these complexes can then enter the cells through endocytosis and phagocytosis and translocate to the nucleus. Finally, biological methods allow virus or transposon-mediated transfection. Depending on the transfection strategy, the target DNA or RNA can either be integrated into the genome and henceforth be expressed stably across generations / cell cycles or in case of a transient transfection be expressed over a limited period of time. A detailed review about transfection methods can be found in [47].

1.5.3 Isogenic Cell Lines

A particularly useful technique in functional genomics is targeted transfection, in which a cell line receives an acceptor vector that is integrated into the genome. The role of the acceptor vector is to facilitate site-specific recombination. To this end, a secondary vector system, which carries a gene or miRNA, is integrated at the acceptor site. Different types of secondary vector systems allow for genes to be expressed in a stable or inducible fashion. This allows researchers to generate a panel of isogenic cell lines that differ only by the gene of interest, which makes them ideal for studying the effect and function of a gene and for drug screening in general [48]. In principle, two different acceptor vectors can be used to create double recombinants with two different genes. These are particularly suitable for studying scenarios of synthetic lethality.

1.5.4 Sample Management

In the past, functional genomics tools have already proven essential for the discovery and characterization of potential oncogenes, e.g. genes that are known to cause cancer, and
potential tumor suppressors [49]. It is clear that large-scale functional genomics studies accumulate not only a significant amount of data, but, independent of the techniques employed, lead to the creation of several thousands of samples representing various stages of the experiment. For example, the maintenance of a vector library for all human genes would already amount to more than 20,000 samples. If a subset of these genes was to be used for stable transfection of a panel of different cell lines, the number of samples would already increase dramatically. It is therefore imperative for any high-throughput functional genomics laboratory to establish and maintain a laboratory information management system (LIMS) to keep track of these samples and the associated experimental data. OpenLab Framework (OLF), which is a novel LIMS system tailored towards high-throughput functional genomics, was implemented during this thesis and is presented in Chapter 3.

1.5.5 Assay Types

Subsequent to the selection and application of a genome perturbation method, a large number of possible experimental assays are available for functional characterization of the perturbed gene. These assays can be divided into the observations of signature changes and observation of phenotypical changes. The most common type of signature data is gene expression data, which allows to compare the relative abundance of mRNA levels between samples or conditions. Originally, gene expression was measured with microarray technology, in which oligonucleotides are immobilized on a glass slide and serve as bait for the mRNA. However, this technology is slowly being replaced by NGS, which also allows measuring mRNA levels in so-called RNA-seq [50]. Using the same techniques, the expression of miRNAs or lncRNAs can also be determined. Moreover, variations of these methods enable researchers to obtain epigenetic signature information by measuring DNA methylation. Finally, a protein expression signature can be established using RPPAs (Chapter 1.7) or mass spectrometry. Phenotypic assays can complement signature information by providing additional information on the capability of cells to proliferate, to change appearance or to cause metastasis, for instance. Phenotypic assays are suitable for HTS.

1.6 High-Throughput Screening

In 1999, shortly before the completion of the human genome project, Evans et al. pointed out that the combination of functional genomics and HTS was expected to drive future
pharmacological studies [51]. This field, also called pharmacogenomics, aims at unraveling the underlying mechanisms of drug action, which is a crucial step for the advancement of precision medicine in cancer. Obviously, a drug utilizing specific genetic traits is of no use if those traits are not found in a particular tumor. Therefore, to know if a drug can be effective, the drug targets need to be identified. Most drugs, however, act through more than one gene, making such studies challenging and highlighting the need for high-throughput approaches that can also deal with polygenic scenarios. Moreover, \textit{LOF} studies require large-scale experimental setups to sequentially test the effect of silencing genes on a genome-wide level.

1.6.1 Automation and Miniaturization

Formerly applied exclusively by pharmaceutical companies, industrial-scale \textit{HTS} is nowadays entering more and more academic institutions, which often seek to extend the application beyond traditional chemical compound screening. \textit{HTS} is generally characterized by a high degree of automation, which allows several thousands of experiments to be performed each day. This is achieved through miniaturization as well as through the extensive use of robotics. Experiments are typically performed in microtiter plates carrying between 96 and 1536 wells. A robotic station for high-throughput drug screening usually consists of an assembly of machines such as plate hotels, incubators, freezers, pipetting stations and instruments for various readouts, which are fully integrated through conveyor belts and robotic arms to achieve full automation.

1.6.2 Types of Readouts

The result of an \textit{HTS} experiment is typically captured with plate readers, which are used to quantify the outcome of an experiment via absorbance, fluorescence, or luminescence. These instruments are often capable of multiple readouts on several wave lengths, allowing for several parameters of an experiment to be measured or to conduct several experiments in parallel. Typical applications are assays that reflect cell viability or metabolic activity. The focus on the numeric output of a plate reader distinguishes \textit{HTS} from \textit{high-content screening (HCS)}, in which automated microscopes are used for image acquisition. For a more detailed review of the development of the \textit{HTS} field see [52].
1.6.3 Types of High-Throughput Drug Target Screens

In most HTS studies, cultured cell lines serve as an in vivo model system. HTS was originally developed by the pharmaceutical industry to be able to test large libraries of small molecules and chemical compounds for drug discovery. Nowadays, many HTS studies utilize libraries of, for instance, siRNAs, small hairpin RNAs (shRNAs), sgRNAs or miRNA inhibitors / mimics to study the effect of a large number of genes up to the genome-wide level. Even though the latter represents a more controlled and targeted approach, chemical compound screening is still very relevant since small molecules can in many cases be engineered into effective drugs, while reagents targeting genes still suffer from the lack of an efficient and specific drug delivery system that can be considered safe in humans [53].

Nevertheless, gene-targeting LOF screens are effective tools for identifying putative drug targets irrespective of how these can later be used for successful treatment. The interrogation of two or more cell lines with the same screening library allows researchers to establish not only vulnerabilities of a single cell-line, but to focus on cell line specific vulnerabilities. The idea is that one cell line serves as a baseline, i.e. genes that prove lethal here are not disease-specific but are generally important for survival and can be disregarded. In this way, differential screening between several cell lines can be helpful to uncover potential scenarios for synthetic lethality [54]. A gene knock-down or knock-out that is lethal in only one of two or more cell lines might point to a synthetic lethal target gene. Possible examples of differential screens include comparisons between cancerous and non-cancerous cell lines, between cell lines representing different molecular subtypes or between cell lines enriched in CSCs and cell lines that consist mainly or exclusively of bulk tumor cells. Due to its versatility, the same HTS platform can also be used to study efficacy of a specific drug on a large number of samples or to test the efficacy of novel drug delivery systems.

1.6.4 Sample Management

Irrespective of the type of HTS experiment, it is clear that sample management is a key issue in the field. Hundreds or even thousands of individual plates need to be tracked efficiently, together with detailed sample information for up to 1536 individual experiments per plate. Moreover, the sample information needs to be linked with one or several types of measurements obtained, e.g. through plate readouts. To deal with these challenges, Sample Management and Visual Analysis of HTS (SAVANAH), was developed (Chapter 5).
1.6.5 Data Analysis

*HTS* data generally suffers from variation caused by, for instance, plate, batch, library, and positional effects [55, 56]. The robust identification of putative hits, e.g. samples with a desired phenotype, from a single screen or the comparison of several independent screens thus crucially depends on normalizing these data appropriately. Unfortunately, there is currently no standard for processing *HTS* data, since different normalization strategies can lead to largely varying results. The inherent problem is that different data sets suffer from different types of variation to a varying degree. Moreover, control samples that are utilized in some normalization methods do not always perform as expected. Therefore, no single normalization method is ideally suited for all data sets. This poses a significant hurdle for the analysis of *HTS* data in general and for secondary and comparative analyses in particular [57]. Additionally, advanced normalization and hit detection strategies can be difficult to apply due to a lack of suitable analysis software. To mitigate this, an intuitive and user-friendly tool named *RNAi* and Compound Screen Evaluation (*RNAice*) was implemented (Chapter 6).

The versatility of a *HTS* platform makes it indispensable for drug target discovery. However, even though it is in principle possible to employ several readout strategies in parallel, the amount of information available from a single experiment is limited. Furthermore, *HTS* experiments cause considerable costs and experimental efforts, such that it is desirable to reuse the treated samples for additional high-throughput experiments. One possibility to gain additional information on the protein level is presented in the following chapter.

1.7 Reverse-Phase Protein Arrays

*Reverse-phase protein arrays* (*RPPAs*) are typically nitrocellulose covered glass slides on which crude lysates of tissue samples or treated cell lines are spotted. Each single slide can carry several thousand spots. Notably, only very small amounts of lysate in the range of a few cell equivalents are required for each spot. Consequently, several hundred slide copies can be created at minimal sample consumption, each of which can be interrogated with a different protein-specific antibody. This allows high-throughput measurement of the relative abundance of proteins in up to several thousand samples. Parallel processing of large sample numbers discerns *RPPAs* from forward phase arrays, where probes are immobilized on a slide, and mass spectrometry, which are both suitable methods for the analysis of many proteins in small sample numbers. Due to the support of large sample numbers, *RPPAs* are well suited as secondary readout for *HTS*, since they can
significantly increase the amount of information gained from each single experiment. In cancer research, RPPAs allow to measure the abundance of known oncogenes on the protein level, establishing a link between the observed effects of silencing a gene and known mechanisms of cancer biology. Measuring these effects on the protein level is a huge advantage over detection methods that measure the mRNA level, since mRNA levels do not necessarily reflect actual protein amounts.

The field of RPPAs has shown steady growth since its introduction in 2001 [58]. Several studies have applied this technology successfully to protein and signaling pathway analyses in cancer [59–62], as well as for cancer classification in general, cancer subtype classification and prognosis of disease progression [63–66]. The relevance of RPPA data for multi-OMICS and high-throughput projects is also highlighted by its inclusion into The Cancer Genome Atlas (TCGA) [67], which already lead to a large pan-cancer study using 181 antibodies for 128 different proteins in 3,467 samples [68].

Experimental challenges involved in this technology, such as antibody selection, sample preparation and optimization of staining conditions have been addressed successfully in the past [69–73]. However, a comprehensive solution addressing bioinformatics challenges emerging from the RPPA technology was missing.

1.7.1 Sample Management and Data Analysis

Using a state-of-the-art spotting device, a single RPPA slide of approximately 12 cm² can easily carry up to approximately 15,000 spots. Each of these spots corresponds to a sample that needs to be tracked back to one of several microtiter plates that serve as sample source for the spotter. In a scenario where RPPAs are used as a secondary read-out, these plates typically originate directly or, if a reformatting step becomes necessary, indirectly from an HTS experiment. Samples are often spotted in varying numbers of depositions and in a dilution series. The various combinations of dilutions and depositions guarantee that a broad dynamic range can be covered for signal detection. This, however, also increases the complexity of the experiment tremendously. Moreover, even single experiments often span several slides, where each slide is spotted in dozens of different copies. This is necessary, since each copy can be stained with a different antibody to measure the abundance of a large number of different proteins. Antibodies can be produced that recognize specific isoforms or post-transcriptional modifications, allowing more fine grained studies. Thus, the number of samples that need to be tracked on RPPAs virtually explodes in a high-throughput setting. This demands for a sample tracking solution tailored towards the characteristics of RPPAs. Due to the fact that
samples are spotted in a dilution series, RPPAs have unique requirements for data analysis. Furthermore, the fact that each protein is measured on a different slide with a different antibody leads to different signal distributions, creating additional challenges for comparing samples across different proteins. A web application tailored towards the specific needs of RPPA sample management and data analysis was lacking in the field [74]. This was the motivation to develop Microarray R-based Analysis of Complex Lysate Experiments (MIRACLE), which addresses both sample management and analysis challenges for customized arrays in general and RPPAs in particular (Chapter 7).

In conclusion, RPPA technology is an appropriate end-point for HTS experiments and contributes essential knowledge for characterizing putative drug targets. The combination of functional genomics, HTS and RPPA therefore constitutes an efficient drug discovery platform.

1.8 An Integrated Functional Genomics Platform for Drug Discovery

High-throughput functional genomics studies combine next-generation cell manipulation techniques with the power of a HTS screening platform. This allows large-scale drug target discovery experiments, aimed at, for instance, identifying disease specific vulnerabilities. Adding a secondary readout at the protein level strengthens this approach considerably in enabling a better characterization of the identified drug targets in the context of established biological knowledge. This constitutes a drug target discovery platform that has the potential to deal with the complexity of multi-factorial diseases and polygenic scenarios such as that of synthetic lethality (Figure 1.5A). Such a platform is thus ideally suited to identify cancer vulnerabilities in general, as well as to discover genes or miRNAs that may serve as a basis for developing drugs targeting, for instance, CSCs with high specificity. Fully harnessing the power of this approach requires not only efficient bioinformatics tools that deal with the specific technological and analysis challenges of functional genomics, HTS and RPPA, but also the tight integration of these tools. This enables researchers to efficiently track large scale experiments across various high-throughput technologies, even though the sample complexity increases in each step. Most importantly, the accumulated information captured by this platform establishes a broad view on gene and miRNA function. Combined with other systems biology resources such as protein-protein interaction networks, these data permit investigating bio-medical questions on a genome-wide level as was already demonstrated in previous studies [75, 76]. Such integrated approaches hold the potential to shed light onto open questions, as, for example, the origin of CSCs and their role in treatment resistance, metastasis and in the emergence of the different cancer subtypes. Moreover,
this can lead to a better understanding of disease biology, which can in turn be exploited to make future drug screening efforts more focused and efficient [77].

**Figure 1.5:** A: Setup of the Drug Discovery Pipeline proposed in this thesis. Existing high-throughput data and biological knowledge can be used to generate testable hypotheses. Functional genomics methods can subsequently be used to design experiments based on gain-of-function or loss-of-function. These alterations can be tested in high-throughput screening setups, which allow drug target discovery at the genome-wide level. Potential drug targets can be characterized at the proteomics level through RPPA technology. This generates new knowledge that again serves as basis for the design of further experiments. B: The thesis structure is inspired by the drug discovery platform. Chapter 2 illustrates how publicly available OMICS data can be used to generate leads for high-throughput experiments. Chapter 3-7 present software tools that solve bioinformatics challenges inherent to the respective domains of the platform.

### 1.9 Thesis Structure

The thesis structure shown in Figure 1.5B is based on the drug target discovery platform introduced above.

- **Chapter 1:** The current chapter provides a general introduction to high-throughput functional genomics, HTS, as well as the RPPA technology.
Chapter 2: A major application of this platform is to widen our knowledge of the emergence of breast cancer and its various molecular subtypes. Therefore, an initial study investigates the issue of correctly classifying breast cancer subtypes from sample data by combining two prominent and widely available types of OMICS data, namely gene expression and DNA methylation, in a statistical learning approach.

Chapter 3: In functional genomics research several hundreds or thousands of samples, such as vector constructs or genetically engineered cell lines, need to be tracked efficiently. Due to a lack for a suitable LIMS for functional genomics research, we developed OpenLabFramework (OLF), a modular and extensible system suitable for functional genomics and other research domains.

Chapter 4: Here, OpenLabNotes, an extension to OLF that allows it to also serve as an electronic laboratory notebook (ELN), is presented.

Chapter 5: This chapter introduces Sample Management and Visual Analysis of HTS (SAVANAH), a sample management system suited to deal with particular challenges that arise when routinely performing HTS experiments, such as the management of extensive molecular screening libraries.

Chapter 6: SAVANAH is complemented by RNAi and Compound Screen Evaluation (RNAice), which deals with challenges in data normalization, visualization and hit detection in HTS data. Moreover, downstream functional analysis is provided for several types of HTS experiments.

Chapter 7: The drug target discovery platform is completed with MIRACLE, which addresses challenges specific to RPPA experiments. This includes sample management, which is complicated by the complex spotting process and data analysis challenges, such as obtaining a single protein concentration estimate from each dilution series on a slide.

Chapter 8: Details about the implementation of the various tools are presented in this chapter. Common as well as unique elements and software design strategies are pointed out.

Chapter 9: This chapter provides a summary of the efforts described in this thesis, as well as a general discussion and conclusion.

Chapter 10: An outlook on future developments and extensions of this drug discovery platform is described in this chapter.

Appendix A-D: The appendix includes references, lists of figures, tables and abbreviations, as well as supplemental tables (Chapter A), figures (Chapter B) and
information about the availability of the source code, additional documentation, and demo applications (Chapter C), and supplemental methods (Chapter D).
Chapter 2

Classification of Breast Cancer Subtypes by Combining Gene Expression and DNA Methylation Data

Objective of this Chapter

Here, we investigate, if DNA methylation data can contribute positively to breast cancer subtype classification. To this end, the performance of four different machine learning models is compared:

- A control model based on the expression of 50 gold standard genes.
- A model based on gene expression data.
- A model based on DNA methylation data.
- A model based on both types of data.

The correct diagnosis of breast cancer subtypes is of great importance for optimal treatment selection and a crucial first step for the successful application of precision medicine. Discovering and utilizing subtype specific vulnerabilities of cancer, however, requires knowledge of robust biomarkers as well as suitable methods that allow one to quickly assess the subtype of a particular tumor. So far, the differences between breast cancer subtypes were mostly studied based on gene expression data [78], even though there is clear indication for the involvement of epigenetic effects [79]. Therefore, the following study that was published in [80] investigates, if DNA methylation data can make
a positive contribution to the performance of breast cancer subtype classification. To this end, we will first highlight the current state of the art in breast cancer subtyping, before investigating how supervised machine learning models could benefit from DNA methylation data.

### 2.1 State of the Art

In contrast to the subjective histological classification through pathologists, gene expression profiling is considered an objective, accurate and robust alternative [81]. To find a set of genes with prognostic properties, Perou et al. performed a systematic characterization of expression profiles of histologically determined subtypes [10]. Sørlie et al. subsequently identified an “intrinsic” set of 427 genes that was significantly associated with disease outcome [11]. In 2009, Parker et al. suggested a more concise set of 50 genes (referred to as PAM50) with good prognostic performance that currently serves as a gold standard for subtype classification [82]. In recent years, various supervised and unsupervised machine learning methods were applied to extract a subset of genes allowing for robust classification of subtypes, including support vector machines [83] and random forests [84]. Furthermore, Daemen et al. exposed a panel of 70 breast cancer cell lines to 90 different therapeutic reagents with the goal to identify prognostic markers that would allow for predicting treatment response using copy number aberration, mutation, gene and protein expression, as well as methylation data. They conclude that no single data set was optimal (25% success rate for transcription data) for delivering prediction markers, emphasizing that multiple data types should be used together (65% success rate).

Online resources such as The Cancer Genome Atlas (TCGA) [67] provide open access to multiple types of breast cancer related OMICS data. This suggests to combine the prognostic potential of gene expression data with data from other OMICS technologies. In this thesis, the focus was on DNA methylation data, which has been shown to play a major role in many cellular processes and in cancer development [85]. Several studies have already shown that random forest models based on gene expression profiles can be used for successful breast cancer subtype classification [78, 84, 86]. Since DNA methylation patterns also differ for breast cancer subtypes [79, 87] and since they can significantly alter the gene expression dynamics [85], it can be expected that DNA methylation data can be applied similarly. Moreover, it can be hypothesized that an integrated model, using DNA methylation and gene expression profiles alike, is superior to the individual models.
2.2 Material & Methods

2.2.1 Data

Gene expression and DNA methylation data were downloaded from TCGA [67] in processed and normalized form. Along with these data, TCGA provided a subtype classification of all gene expression samples via the gold standard PAM50. The gene expression data set contained samples of 547 breast cancer patients. Three samples were marked as metastatic and removed. We could match all but one of the remaining 544 samples to the methylation data set, resulting in a total of 543 samples for analysis. It should be noted that of the 30 samples recognized as normal by PAM50, only 22 were actual non-tumor samples according to the sample identifier used by TCGA.

2.2.2 Classification

2.2.2.1 Random Forest and Bootstrapping

We applied the varSelRF R package [78] to perform random forest classification on (1) the gene expression dataset (gene expression model), (2) the DNA methylation data set (methylation model), (3) a combination of both datasets (combined model) and (4) a reference model, in which we applied random forest using only the 50 genes used in the PAM50 classifier (control model). The combined feature space was created by appending all feature columns of the methylation to the gene expression feature matrix, allowing the random forest method to use both sources for feature selection (Figure 2.1).

Recursive Feature Elimination:

For each model, we applied varSelRF with default settings. Initially, 5000 trees were constructed, in order to remove the bulk of features that were not relevant for classification. For each tree the square root of the number of features of the current feature space were selected for construction. Subsequently, the feature set was further reduced in several additional random forest runs with 2000 trees each, while the set size was continuously reduced by dropping the 20% least important features. This so-called recursive feature elimination was repeated until a tree with only two splits was left. Afterwards, the model with the lowest out-of-bag error (OOB) was returned as solution.

Bootstrapping:

In order to assess how results between several independent runs agreed and how much our models suffered from overfitting, we applied the .632 bootstrap method [88]. This
method is similar to cross validation, since only one part of the data set is used for training the model, while the other part is kept for model validation. In contrast to cross validation, however, the .632 bootstrap method creates datasets of the same size as the original dataset through sampling with replacement and is therefore sometimes also referred to as smoothed cross validation. Following 10 iterations of bootstrapping, we calculated the .632 bootstrap error, which is defined as the weighted average of (1) the classification error on the training data and (2) the so-called leave-one-out bootstrap error, where samples that were not part of the training data (the ones “left out”) were used to validate the model [78].

### 2.2.2.2 Feature Importance

The most significant features were extracted through sorting of all feature lists for the mean decrease of the gini index or gini coefficient, which is an established measure of feature importance in random forests [89]. In a binary decision tree $T$ for two classes $A$ and $B$, the gini index $i(n)$ is defined for any given node $n \in T$ as follows:
where \( p(A|n) \) is the proportion of cases of class \( A \) in node \( n \) and \( p(B|n) \) is the proportion of cases of class \( B \) in node \( n \) respectively. The gini index is thus a measure of node impurity. It can be generalized for an arbitrary number of classes \( j \)

\[
i(n) = \sum_j (p(j|n) \cdot (1 - p(j|n)))
\]

(2.3)

The decrease in gini index in a split at node \( n \) in a decision tree is defined recursively as a function of its own gini index and that of its children \( k \), i.e.

\[
\Delta i(n) = i(n) - \sum_k (p(k)i(k))
\]

(2.4)

where \( p(k) \) is the proportion of samples sent to child \( k \) and \( i(k) \) the gini index of child \( k \) respectively. Applied to a random forest, this measure is averaged across all decision trees to provide a summarized measure of the importance of a given feature. The decrease in gini index is large for features that separate samples well into the pre-defined classes \( j \) and small otherwise.

### 2.2.2.3 Model Evaluation

Traditionally, the classification performance of random forests is assessed through receiver operator characteristics (ROC) curves and summarized by the area under the ROC curve (AUC). This solution, however, is only applicable to binary classification, since the two dimensions of such a plot fail to capture more than two classes. As an alternative, we calculated the average AUC (AAUC), sometimes referred to as multiclass AUC, based on all pair-wise class combinations as described by Hand and Till [90].

Bootstrapping allowed us to evaluate the performance of the different models via the .632 bootstrap error. However, since these models were based on part of the data only, we applied a final random forest run with 5000 trees using only the previously selected features of the bootstrap solution, but the complete dataset. This allowed us to assess the overall classification error of each model by computing the confusion matrix and the AAUC.
2.2.3 Evaluation of Feature Lists

We evaluated how the feature lists overlap with each other and with a list of 373 known breast cancer genes downloaded (01/17/2014) from the Network of Cancer Genes (NCG) [49], as well as with the intrinsic gene list of 1918 genes that have been aggregated by Parker et al. during their effort to develop the PAM50 classifier [82]. In order to deal with synonymous gene symbols, we parsed all symbols to their unique entrez identifiers.

2.3 Results

Following our hypothesis that epigenetic data can make a valuable contribution to improving the classification of breast cancer subtypes, we compared different random forest models. In the following, we show how these models performed in terms of accuracy, which features made the most significant contribution to the classification, and how the selected features overlap with known breast cancer genes.

2.3.1 Classification Performance of the Different Models

As shown in Figure 2.2, the gene expression model performed best with a very low bootstrap error of less than 10% and an AAUC of close to 100%, which was also the case for the combined model and the control model. The methylation model performed slightly worse, achieving a bootstrap error of 20% and an AAUC of 88%.

![Figure 2.2: The .632 bootstrap error over 10 iterations (left), AAUC (middle), and the distribution of the number of variables across bootstrap iterations (right) after applying random forests to the different feature matrices. For comparison, the AAUC of the control model is also shown.](image)
As shown in Figure 2.3, all random forests models converged on their optimal classification performance within the first 1000 trees. The *gene expression* and the *combined model* could separate all subtypes with a classification error varying between 1 and approximately 30%. The best classification performance was achieved for the basal and luminal A subtype, whereas the worst performance was found for the HER2 subtype and samples labeled as normal. The *methylation model* yielded similar results with generally higher error rates between 6 and 50%.

**Figure 2.3:** The classification and out of bag error (OOB) of each subtype for the control model (A), gene expression model (B), methylation model (C) and the combined model (D) depending on the number of trees already created by the random forest algorithm.

In contrast to the other two models, the *methylation model* showed the worst classification performance for the luminal B instead of the HER2 subtype. This is illustrated in more detail in the confusion matrices (Table 2.1), which shows which pairs of subtypes were mainly confused by each model. All models confused the two luminal types A and B. HER2 was mostly confused with the luminal B subtype.

### 2.3.2 Analysis of Misclassification by the Control Model

A total of 30 samples were labeled as normal by the PAM50 classifier. Our results (Table 2.1) show, however, that 8 of these samples had been assigned to different breast cancer subtypes, leading to a high classification error for the normal class. A possible explanation for this was offered by the *TCGA* barcodes linked to these samples, which indeed identified them as originating from tumors. We therefore repeated the analysis
excluding these samples. This led to comparable results as illustrated in Supplemental Figures B.4 and B.5. The only difference we found concerned the number of selected features, which was generally higher in the gene expression models when excluding these samples.

### 2.3.3 Most Important Features

Figure 2.4 depicts, for each of the models, the features with the highest mean decrease in gini index. In other words, these were the 15 features with the highest importance for the classification. Among the selected features stemming from gene expression data
we find many known cancer genes. The top four hits overlap between the control model, the gene expression model, and the combined model. As expected, we identified the estrogen receptor (ESR1), which is the most relevant marker for breast cancer subtyping in histochemistry, as one of the top features. The other top features are FOXA1, which has been identified as a tumor suppressor [91] and MLPH, which is not known to be associated with breast cancer, but has an experimentally verified interaction to RAB27A, which was shown to promote proliferation in human glioma cells [92]. Furthermore, we identified FOXC1 as a top feature, which is of the same family of transcription factors as FOXA1 and which was shown to play a role in NF-κB signaling in basal breast cancer cells [93].

The top hits of the methylation model showed no overlap with the gene expression derived features. The top five features include Dynamin-3 (DNM3), which is a novel tumor suppressor candidate in hepatocellular carcinoma [94], TNFRSF10A, which is a member of the tumor necrosis factor receptor superfamily, Sorcin (SRI), which several studies could connect to multi-drug resistance in cancer [95], TBX19 (also known as TPIT), which has not yet been associated with cancer, and TFF1, which is known to be regulated by DNA methylation and which is a predictive factor for poor survival in gastric cancer [96]. The trefoil factor family is characterized by a 40 amino acid motif and

![Figure 2.4: The 15 most important features determined by the mean decrease in gini index for models based on control model (A), gene expression model (B), methylation model (C) and combined model (D).]
interestingly TFF1 and another member of this family, TFF3, are found in the feature list of the combined model. TFF3 expression is positively correlated with the status of the estrogen receptor in adenocarcinoma [97] and TFF3 is furthermore associated with breast cancer invasion and metastasis [98], as well as treatment resistance [99]. For each model, the importance of the top 50 genes is shown in Supplemental Figures B.1 to B.3.

The number of selected features varied across bootstrap runs (Figure 2.2). The combined model selected the highest number of features, followed by the methylation and the gene expression models. The superior performance of the gene expression model is also reflected in the final feature list of the combined model, which consists almost exclusively of features derived from the gene expression part of the matrix. Furthermore, these feature lists show very strong overlap and consequently, the two models perform equally. Even though the feature lists across different runs were not identical and varied in size, the same features were always found at the top in slightly changing order.

### 2.3.4 Overlap of Feature Lists

Figure 2.5 depicts how the different features selected by the different methods overlap with each other and with the PAM50 gene list. Only 13 of the 38 features from the methylation model were reported by the combined model. In contrast, all 53 features of the gene expression model were included. 32 PAM50 genes were included in the combined model, out of which 14 were also found in the gene expression model. Only a single gene, namely MIA, was found by the methylation model, as well as in the PAM50 list.

![Venn diagram showing the overlap of the feature lists selected by each of the random forest models and the PAM50 genes that serve as gold standard.](image)

A comparison of the methylation and combined model with NCG and intrinsic lists (Figure 2.6) reveals that only 10 of the 278 genes in the combined model, and none of 38 features of the methylation model were listed in the NCG list. 127 of the genes of the combined model and 8 genes from the methylation model could also be found in the
intrinsic gene list by Parker et al. The intrinsic gene list and the NCG query list show an overlap of 48 genes.

![Figure 2.6: Venn diagram showing the overlap of the features selected by the combined model with a list of known breast cancer genes (downloaded 01/17/2014) from the Network of Cancer Genes [49] and with the list of intrinsic genes reported by Parker et al. [82].](image)

### 2.4 Discussion & Conclusion

Both, gene expression and DNA methylation are typically measured through microarrays comprising thousands of probes. Most of the probes, however, can be considered as background, since the fold change of the represented genes is negligible after normalization [100]. One of the most crucial tasks in microarray data processing is thus identifying probes that show significant differential signal between different conditions or groups, e.g. between different breast cancer subtypes. Machine learning algorithms, such as random forests, are able to handle the noise in the data while providing both, good accuracy and a manageable run time [86]. The original implementation of random forest by Breiman [89] struggles with microarray data, since a large number of probes will make a small but negligible contribution to a successful classification. As a result, each new run of the algorithm will lead to large feature lists with little overlap. Using the varselRF R package [78] allowed us to address this problem by recursively removing features with low significance from the feature list, such that subsequent trees had a higher likelihood of incorporating significant features. In this way, we could show that random forests are suitable for extracting relevant features from both, gene expression and DNA methylation data.

In our analysis, the performance of the gene expression model was superior in consistency, which was measured through .632 bootstrapping, as well as in accuracy, which was determined through an AAUC and classification error rates. Consequently, the feature elimination process removed most of the methylation derived features in the combined
Both models showed a similar classification error and AAUC as the control model.

It should be noted that the control model does not achieve a perfect classification, due to the fact that the PAM50 classification is based on the method of shrunken centroids, while we use the same 50 genes as input for creating a random forest model. Furthermore, class labels may flip even in the shrunken centroid method, when samples have the same correlation distance to two different centroids, such as luminal A and luminal B [82].

The generally high classification error with the HER2 subtype can be explained by the fact that the training data was highly unbalanced in favor of the basal and luminal subtypes. Another explanation might be that even though this subtype is identified through HER2 (also known as ERBB2) expression, the combined model did not select this gene, but interestingly EGFR, ERBB3 and ERBB4, which belong to the same protein family as HER2.

The imbalance of the training data might also offer an explanation for the classification error found for samples labeled as normal. Here, we could, however, determine via the TCGA provided sample information that 8 out of 30 samples were in fact originating from tumor tissue. Removing these samples prior to the analysis lead to comparable results with a higher number of features selected for the gene expression model, indicating that these samples might in fact have perturbed the feature selection. Finally, the confusion matrices (Table 2.1) clearly show that all models had difficulties separating the two luminal subtypes, indicating that they are highly similar on a transcriptomic and epigenetic level.

All feature lists obtained in this analysis showed very little overlap with the list of known breast cancer genes derived from the NCG. This agrees with previous findings that gene selection remains a challenge in which different, or even the same methods, can lead to highly disparate results [86]. This is also emphasized by the high variability in the number of selected features in all models, reaching from as few as 8 up to more than 300 features (Figure 2.2).

The relatively large overlap of our combined model with the intrinsic list of Parker et al. and the PAM50 gene list can be explained through the fact that the TCGA classification of the samples was performed using PAM50. This, however, is an inherent problem of supervised learning that we cannot easily overcome. Assumptions about the current classification system have already been challenged by unsupervised learning methods that have extracted clusters that only partially agree with known breast cancer subtypes.

In the TCGA publication on breast cancer subtypes itself [67], 13 distinct breast cancer subtypes were found on gene expression data using a method called SigClust [101], while
for the methylation data, the authors were able to extract five clusters using a recursively partitioned mixture model \cite{102}. As depicted in the supplemental material of \cite{67}, these clusters do not agree with the classical subtype scheme. This highlights that clustering of methylation data might reveal other differences than the ones that can be explained by the established subtypes. Our analysis showed, however, that at least some of the features can be used for successfully classifying tumor samples according to clinically relevant subtypes.

In conclusion, gene expression data appeared to be superior to DNA methylation data for breast cancer subtype classification. It remains unclear, however, if using the PAM50 gene expression discriminator as the gold standard still allowed a fair comparison. Furthermore, the imbalance in the training data made it difficult to identify features suitable to discriminate HER2 from other subtypes. These issues could in the future be addressed by either generating a more balanced data set or by compensating with class weights.

The small overlap of the feature lists found here and the intrinsic gene list with the NCG query list emphasizes that genes that are known to play an important role in cancer might not necessarily be the ones most suitable for breast cancer subtype classification.

Finally, we expect that further research into the most promising features of the methylation model might lead to a better understanding of breast cancer subtype differences on an epigenetic level. A good example for this is Sorcin, which is not only the most prominent feature in the methylation data, but also appears to play a major role in resistance to cancer treatment. In contrast to the well studied genes found in gene expression data, many of the genes found here, such as TBX19, have not yet been linked to breast cancer and would thus be ideal candidates for functional genomics experiments to study their role in cancer development in the different subtypes and to investigate their potential as drug targets.
Chapter 3

OpenLabFramework: A Laboratory Information Management System with Focus On Functional Genomics

Objective of this Chapter

Development of a modular and extensible laboratory information management system (LIMS) to cover functional genomics workflows, including

- sample tracking in hierarchical storage structures.
- QR barcode label printing and scanning.
- support of mobile devices.
- document management.

In functional genomics, researchers utilize powerful methods to study the effect of knockdown, knock-out or over-expression of a gene to be able to establish its function, its role in a disease, and its suitability as a drug target (Chapter 1.5). With the development of these sophisticated cell manipulation techniques, functional genomics has seen a paradigm shift from small projects involving single or few researchers towards large-scale projects involving several laboratories and often hundreds or thousands of samples. Sample management is therefore a growing issue, especially since most laboratories still attempt to keep track of their samples using spreadsheet tools. A high turn-over of
academic staff coupled with maintenance of individual files that are often locked or outdated, as well as inconsistent nomenclature and labeling, can lead to tedious repetition of previously existing work. The significant amount of time that is often spent on locating samples would be better used for performing experiments. Moreover, expensive storage space is wasted, since samples are often not labeled properly and cannot be identified. Even if a label is given, it usually does not include a standardized minimal amount of information that allows unambiguous identification of the materials or the experiments they were derived from. These issues can be overcome by introducing and maintaining a suitable LIMS. Existing LIMS already cover a large variety of experimental work-flows, mostly focusing on a particular field of research with specific requirements. To be useful across various disciplines, LIMS needs to be implemented in a flexible and extensible way, covering a broad range of possible use cases. In this chapter, we will first identify common requirements for a LIMS system and subsequently, we will analyze to what degree these requirements are already fulfilled by existing solutions. This analysis is followed by a detailed description of OpenLabFramework (OLF), a novel LIMS that was developed during this thesis to better accommodate the needs of the functional genomics community. OLF was previously published in [103], which is the main source of this chapter.

3.1 Requirements

Any LIMS that involves sample management on a large scale should in our opinion fulfill a number of requirements listed in the following as R1-15. These requirements are the basis for a comprehensive comparison with existing solutions in Supplemental Table A.2.

3.1.1 Implementation

A LIMS for an academic environment needs to be open-source (R1), in order to save costs and to allow for adaptation to the specific requirements of a given scientific field and laboratory. Since adaptation can be a difficult and time-consuming task, a LIMS that is modular and extensible by design (R2) would be most appropriate. Although difficult to assess for existing projects, a LIMS should be reliable and its implementation simple. Existing frameworks and software packages that are maintained and tested by a large community are often more reliable than individual solutions and should thus be incorporated.
3.1.2 Data Handling

Dealing with a large number of samples in a library or biobank requires efficient mechanisms for sample management (R3) and physical sample tracking over several hierarchical levels (R4). Since related information and experimental results are usually stored in additional documents, a management system, where files can be linked to an arbitrary number of samples (R5), would be most useful. Another requirement is that raw data previously entered into the system can be exported to various file formats. This requirement is usually met through an integrated reporting mechanism (R6).

3.1.3 Flexibility in Deployment

Academic laboratories are often part of an existing IT infrastructure, but support is in many cases limited, e.g. to a single database management system (DBMS), such as MySQL\(^1\). LIMS deployment should thus be as flexible as possible, not be bound to a specific operating system or DBMS. While the first requirement is fulfilled by all LIMS considered here, multiple database support remains an issue (R7). Furthermore, if a suitable server is not available, deployment locally (R8) or to a cloud service (R9) is advantageous.

3.1.4 User Acceptance and Excess Value

Triplet et al. have identified approachability as a major hurdle in the acceptance of a LIMS [104]. Modern web-technologies allow for a more responsive and intuitive user interface, which in turn improves the user experience and reduces the learning period. User acceptance can also be improved by offering an excess value over traditional spreadsheet tools, for instance by incorporating the use of barcodes (R10), label printing (R11), and mobile devices, such as smartphones (R12). This is particularly useful in case experimental data from experiments can be uploaded to the system. Finally, a crucial requirement for a successful adaptation of a LIMS is good documentation (R13). A further advantage would be the incorporation of data analysis tools directly within the LIMS (R14).

3.1.5 Security

LIMS typically address security concerns by restricting access through secure user logins and different user roles. Security would also be enhanced by audit logging features

---

\(^1\text{https://www.mysql.com/ (access date: 06/01/2015)}\)
(R15), where a version number is added to each database entry. Any change will then result in a copy of the entry with a new version number, so that accidentally overwritten entries can be restored.

The requirements identified here allow assessing how well existing solutions cover the needs of the functional genomics research community.

### 3.2 State of the Art

While a large number of laboratories continue to rely on spreadsheet tools, it is obvious that a powerful LIMS is advantageous in handling the large number of samples that are regularly created in a functional genomics laboratory. This also holds true for most other bio-medical research areas, which has led to the development of countless new LIMS platforms in the past 20 years. Notably, there are a number of commercial providers like Labvantage® that offer adaptable systems for a large number of use cases. The licensing costs for these systems, however, are usually not affordable for a typical mid-sized laboratory. We therefore focus on available open source alternatives that have been developed within the scientific community.

Existing LIMS are often tailored towards specific types of research data, as for instance genotyping [105–107], protein production [108, 109], protein-protein-interaction [110], 2D gel electrophoresis [111], or protein crystallography [112] data. Some generic LIMS target specific laboratory tasks, such as sample management [104, 107, 113], laboratory work-flows and protocols [106, 114–116], documentation, management of lab stocks, or clinical studies [117]. Additional solutions exist for molecular genetics and the creation of vector libraries [118]. There is, however, no dedicated LIMS for handling samples typically generated in functional genomics studies. The complete list of currently available LIMS is provided in Supplemental Table A.1. However, only those LIMS that support sample management were considered for a detailed requirement analysis, which can be found in Supplemental Table A.2.

**Summary:**

None of the existing solutions provides the necessary flexibility to be easily adaptable to new sample types. Moreover, the existing systems have typically very strict requirements for their deployment. Despite their usefulness, barcode generation is only supported by two and label printing by one tool. The possible advantages of incorporating mobile devices in the laboratory work-flow have not been tested in any of the existing LIMS. The lack of flexibility and supported features motivated the development of OLF to
provide a LIMS that is equipped to face the sample management and documentation challenges of the functional genomics community.

3.3 Results

OLF is primarily targeted at advanced sample and storage management in mid-sized laboratories with less than 50 users. It facilitates a seamless integration of virtual and real world storage handling by making use of mobile devices, which are carried by lab personal anyways, in combination with cheap and fully integrated barcode labeling technology. In the following we shed a light on how OLF fulfills the LIMS requirements that we have identified before (R1-R15) (Supplemental Table A.2).

3.3.1 Modularity and Extendibility

OLF was published under an open source license (R1) and, due to its modular structure, it can be adapted to different types of laboratory data and sample types (R2) (Figure 3.1).

![Figure 3.1: OLF is built in a strictly modular fashion. A back-end module provides the basic functionality, including project and user management, as well as base classes for other modules. Additional modules (depicted exemplary as ?) extend the base classes and integrate with existing ones. Finally, the front-end module creates views for all defined content and allows for interaction through a responsive web-interface.](image-url)
3.3.2 Sample Management

The OpenLabGeneTracker module is intended to fulfill requirements specific to the hierarchical organization of genes, gene variants, vector constructs, and genetically engineered cell lines (R3). The organization of these samples is further supported through OLF’s built-in user and project management features.

3.3.3 Sample Storage

The OpenLabStorage module adds options for tracking and organizing samples in a customizable storage infrastructure (R4). This infrastructure is hierarchical, starting from buildings and rooms and ending in individual freezers and storage boxes. Interactive grids help the user to assess the content of a storage box at a glance. Together with OpenLabGeneTracker, samples can be added or removed from storage in an intuitive manner, while providing an overview of remaining copies and related samples.

3.3.4 File Uploads

The OpenLabFileAttachments module allows users to up- and download arbitrary files, enabling users to upload and organize their results and documents. Files are stored with a combination of timestamp and original file name to avoid conflicts arising from identical file names. Files are uploaded to a configurable folder on the server and not to the database itself. They can be linked to an arbitrary number of samples, so that other users can quickly obtain an overview of files relevant to a sample (R5).

3.3.5 Reporting

OLF can export lists of samples to various file formats, including Excel (XLSX), Open Document Spreadsheets (ODS), PDF, and comma separated values (CSV). This feature is currently available for lists of genes, vector constructs, and cell lines. The storage hierarchy and individual boxes can also be exported to a spreadsheet format (R6).

3.3.6 Deployment

OLF is not bound to a specific DBMS and will also work with non-SQL systems, such as MongoDB\(^2\) (R7). OLF is compiled either as WAR file, which is suitable for deployment

\(^2\)https://www.mongodb.org/ (access date: 06/01/2015)
on a large number of Java-based web containers, or as locally executable JAR file, which comes packed with its own web container and file-based SQL solution (R8). Finally, OLF is also suitable for cloud deployment, using, for instance the VMware CloudFoundry service\(^3\) (R9).

### 3.3.7 Barcode and Label Support

The functionality of the OpenLabStorage module is complemented by the OpenLabBarcode module, with which a user can create and print barcode labels (R10,11). These can later be used to locate a sample in OLF by scanning the barcode using a USB-connected scanner or a mobile device (R12). The Barcode module currently requires a connected DYMO\(^\text{®}\) label printer but can be extended in the future to support other devices.

### 3.3.8 Web Application Interface and Documentation

OLF offers a modern web-interface that is clearly organized and intuitive (Figure 3.2), and allows for responsive user interaction. At the top, a navigation menu provides direct access to all features contributed by OLF and its modules. The search functionality allows users to locate required information quickly and conveniently. Samples in OLF can be grouped in projects, which are accessible through a tree structure shown on the left-hand side. Modules can contribute so-called add-ins to OLF. These can be arranged freely by the user in a column on the right-hand side. Add-ins can react to the main content shown in the middle of the screen through an event-based system. An example for this is the barcode add-in, which will offer label printing options when a sample is shown, or show a barcode scanner view otherwise. OLF distinguishes mobile clients from desktop clients. If a mobile device is detected, a different view is shown that is tailored for the small-sized screen and touch-screen interaction (R12). Finally, OLF comes with online documentation that introduces the system to users, administrators, and software developers (R13).

### 3.3.9 Typical Work-Flow

Users can develop effective laboratory work-flows using the sample tracking feature together with barcode labels and mobile devices (Figure 3.3). Only administrators can add master data, such as cell lines or vectors, which can then be utilized by normal users to create samples. OLF validates all user entered data for validity and will, where applicable, provide a list of viable options in form of select boxes. In this way, OLF

\(^3\)http://www.cloudfoundry.com/ (access date: 06/01/2015)
Figure 3.2: The web-interface of OLF is divided into four main parts. (A) The header contains a menu and a search field for navigation. (B) A hierarchical project tree found in the left column can also be used for navigation. (C) The main panel is used to render the actual page. It can be further divided into a central panel (1), where properties can be altered, an object history box (2), an operations box with additional links (3), and at the bottom a set of tabs (4) where related information is available and can be interacted with. (D) Additional interaction possibilities are provided through add-ins that can be customized by each user in the right column.

effectively avoids ambiguity and ensures consistency of sample data. Once a sample, for instance, a vector construct or a cell line, has been created, OLF will offer the possibility to print a barcode label. The user can then apply the label and add the sample to physical, as well as virtual storage. At a later point, the barcode can be used for efficient retrieval and updating of sample information. Moreover, new gene variants and passages can be added with respective new labels and storage locations conveniently. Files and documents can be added to samples and genes, in order to make experimental results and additional information such as related publications, available to other users

3.4 Discussion & Conclusion

Numerous commercial and open-source LIMS exist today. However, since commercial licenses are expensive and lack the possibility to be adapted to specific needs without additional costs, academic laboratories usually focus on finding an open-source solution
Figure 3.3: A typical work-flow in OLF: Initially, the administrators set up master data, such as vectors, cell-lines, medium compositions, as well as the storage infrastructure (*). Users then create projects and link genes to them. Vector clones are created from the genes, which in turn can be used to create cell-line recombinants. OLF supports printing of barcode labels, which can be scanned at a later point to modify existing or to add new samples.

to their sample management issues. Moreover, none of the existing solutions seems optimal for all given tasks (Supplemental Table A.2). For example, none of the tools supports the samples and work-flow typically encountered in a functional genomics laboratory. Particularly support for printing and scanning of barcoded labels would be a desirable feature to increase productivity. Naturally, most LIMS are dedicated to a specific field of research and are thus not generally suited for other fields. Some solutions, on the other hand, focus on certain general aspects of laboratory work, such as sample tracking, protocols, or work-flows. Acknowledging that many LIMS remain limited to their research domain, we created OLF in a strictly modular and extensible fashion, with dedicated modules for sample tracking, file management, and barcoded label printing. We expect that OLF can be adapted to other research fields and biomaterials with minimal developmental effort by implementing a content module similar to OpenLabGeneTracker, which is then complemented by plugging in additional features as needed. Since the successful publication of OLF, the power of this modular approach could already be demonstrated by the addition of OpenLabNotes, which is a module that adds ELN functionality to OLF (Chapter 4). The integration of an ELN into a LIMS is a novelty among open-source ELNs and is expected to further increase productivity by enabling OLF users to maintain all experimental details, including documentation they are legally required to maintain, in a single place.

The introduction of OLF allows controlling sample logistics effectively, which is a particular challenge upon movement, turn-over of lab staff, and improper labeling of samples. OLF may further increase productivity by including modern technologies so far disregarded by most other open-source LIMS, such as printing and reading barcode labels. The high degree of automation and standardization that can be achieved by this may
substantially reduce user-caused errors in sample assignment. A web-layer for mobile devices provides an additional advantage. In this way, samples can now be removed from physical and virtual storage at the same time, thus limiting the risk of forgetting this step after the work with the sample is completed. As illustrated in Figure 3.3, the implementation of OLF in a laboratory environment can lead to a significantly more productive work-flow.

Unlike most LIMS, OLF is not bound to a specific database or web-container. It can be coupled with a large number of database management systems, including non-SQL solutions like MongoDB. If a suitable server is not available, OLF can be installed locally, on a server, or even be deployed to the cloud. This flexibility, together with the online documentation, will reduce technical hurdles in the introduction of OLF to a new laboratory.

Basic reporting capabilities are available, but could be substantially improved by allowing for customized reports, where information from several instances is pooled. This would also help in establishing data analysis directly within OLF. Finally, OLF’s database security would benefit from the introduction of an audit logging feature, which could, for example, prevent sample data from being accidentally deleted.

In conclusion, OLF offers efficient and user-friendly management of sample information and location in the field of high-throughput biology and functional genomics. Being extensible, it can be further adapted to satisfy additional requirements with little developmental effort. An example for this is presented in the following chapter.
Chapter 4

OpenLabNotes: An Electronic Laboratory Notebook Extension for OpenLabFramework

Objective of this Chapter

Development of an electronic laboratory notebook (ELN) extension for OpenLabFramework, including

- a user-friendly document editor.
- content protection through access control and digital signatures.
- links to existing samples and documents.

In laboratory research, experimental work is typically documented in paper laboratory notebooks (PLNs). This does not only help researchers to keep track of the details of their experiments, but also protects their interests, and those of their institutions, in legal disputes. For a long time, electronic data storage was not considered a legally acceptable alternative, since information could be falsified more easily than in PLNs. Therefore, electronic laboratory notebook (ELN) were only considered as a supplement, but not as a substitute for PLNs. However, ELNs have clear advantages compared to their paper-based pendants. Laboratories are open environments and thus documents can get accidentally destroyed, lost, or, in the worst case, stolen. ELNs, on the other hand, are typically paired with an automated and reliable backup system. Time stamps and additional meta information can further contribute to the reliable documentation.
of costly experimental efforts. Moreover, there are immediate advantages in daily use, such as full text search, which allows researchers to retrieve information quickly and to share documents with their colleagues. In this way, ELNs can significantly increase productivity and make it easier to keep track of experimental details despite of frequent turn-over in laboratory staff [119]. Thus, when the U.S. Federal Drug Administration introduced a guideline for electronic information storage (CFR21 part 111), ELNs slowly began to replace PLNs in many laboratories. This leads to a list of general requirements an ELN should fulfill, which are denoted in the following as R1-18 and which can also be found in Supplemental Table A.3.

4.1 Requirements

In order to follow the regulations, an ELN needs to guarantee that documents are safely stored. The history of a document, such as the creation date and the author, must be traceable (R1). To have credibility in a law case, documents must be protected from changes. This implies that a document in the ELN should be signed by its author (R2). Ideally, it is possible for a colleague or a supervisor to countersign a document to further increase its credibility (R3). The latter is particularly important when it comes to patenting, since a countersigned document that includes a timestamp can be considered strong legal proof for an invention. In a typical software system, a user proves his identity by securely logging into a system with a unique username and a secret password. However, instead of merely marking a document as signed by a logged in user, a so-called digital signature should be used to prevent scientific fraud not only on the application, but also on the database level. Digital signatures rely on a pair of asymmetric keys, a private key, to sign the document, and a public key, to validate the document [120].

To enable researchers to collaborate effectively, an ELN should enable its users to organize documents in notebooks (R4) and projects (R5), as well as to share them with other users (R6-7). Moreover, it would be advantageous if users could comment on or to annotate documents (R8) and upload additional files (R9). A user-friendly interface (R10) and a rich text editor (R11-12) allows users to document their experimental results conveniently. Various sort and filter options guarantee that crucial information can be quickly located (R13-15). Export to local file formats should be possible (R16-17).

ELNs focus on the thorough documentation of experimental procedures, including applied methods, consumables, as well as experimental results. Another type of electronic

\footnote{http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/CFRSearch.cfm?CFRPart=11 (access date: 06/01/2015)}
system that can increase productivity in laboratories is the laboratory information management system (LIMS), which is mostly concerned with sample management. Integrating an ELN with a LIMS enables researchers to link experimental documentation directly to the samples that were part of the experiment (R18). This is particularly useful, if an experiment is to be repeated or continued by another researcher not familiar with the experiment.

4.2 State of the Art

A large number of commercial ELNs are already on the market [121]. However the costs for these systems can typically not be covered by small to mid-sized laboratories. Free alternatives range from general note-keeping tools to dedicated ELNs. A comprehensive list of commercial and free ELNs can be found online\(^2\). The majority of the free systems, however, do not sufficiently protect the documents to prevent scientific fraud. Moreover, many of these are web applications that are only accessible online and cannot be deployed in a local network. Thus, researchers using these systems might be in violation of strict data protection rules found in many academic institutions. Voegele et al. found that none of the existing ELNs could sufficiently fulfill the requirements found in a typical laboratory and thus developed a new open-source ELN based on WordPress [122]. Their IARC ELN prevents scientific fraud on the application level by user authentication and on the database level by securing and monitoring database access. The lack of digital signatures in this system, however, may pose a problem in case of patenting, where it is advantageous if a document describing an experiment that led to an invention is actively signed and protected by an officially recognized digital signature algorithm. Finally, none of the available open-source ELNs, including the IARC ELN, feature the integration with a LIMS. We thus sought to extend OLF to close this gap. A detailed comparison between the features of the IARC ELN and OpenLabNotes can be found in Supplemental Table A.3.

4.3 Results

We realized that experimental documentation in OLF (Chapter 3) through uploading additional files or through using comment fields was not a convenient solution. More importantly, this solution fell short of the legal requirements an ELN is expected to fulfill. OpenLabNotes is a module for OLF that closes this gap by implementing ELN functionality (Figure 4.1). This module was initially developed during an individual

\(^2\)http://www.atriumresearch.com/html/eln.htm (access date: 06/01/2015)
study activity by Michael Franz\textsuperscript{3} and later extended by the author. In OpenLabNotes, a rich text editor allows documents, which are called notes, to be created with the various formatting options that a user typically expects from a modern text editor. Images can be placed within the text and will be stored as part of the note. Each note can be edited until it is finalized by the user (draft status). In this case, the note, as well as a timestamp, are digitally signed with the private key and password of the user (final status). When signing a note, the user needs to appoint a supervisor, which can then countersign it (signed status). These signatures are subsequently validated each time a note is shown to a user. In order to document the origin and current location of material used in an experiment, notes can be linked to arbitrary samples in OLF. Moreover, notes can be attached to specific projects or organized in notebooks, which are user-defined collections of notes. The documentation of an experiment can be further complemented by uploading additional files. Frequently, notes need to be accessible to researchers other than the author or the assigned supervisor. To this end, each user with access to a note can also grant access to other users. To retrieve notes, several types of lists are available, including a list of authored notes, a list of accessible notes, a list of notes that were signed in the role of a supervisor and most importantly, a list of notes a supervisor still has to sign. Lists can be filtered by author, data, and project. Each list includes a preview of the notes and can also be exported to various formats, including XLSX, ODT, DOC, and PDF. When browsing samples, a list of linked notes is shown in a dedicated tab. Most importantly, notes can be retrieved through the search functionality, which extends to entire text of the note. Finally, notes can be printed or exported to either PDF or DOCX format.

4.4 Discussion & Conclusion

ELNs revolutionize the way in which experimental notes are kept by making them more secure, accessible and shareable. The requirements to data safety make ELNs more reliable than their paper based pendants and it can be expected that researchers will be required to use ELNs in the future \cite{119}. The choice of an appropriate solution, however, is difficult. The market offers a wide variety of products resulting in a large choice of attractive and reliable products that are mostly superior to free alternatives. As a disadvantage, these solutions are often not affordable to laboratories. Moreover, it is unclear if a company will endure in the competition and still be able to provide support in a long term perspective. It might therefore become necessary to migrate from one ELN to another, which is complicated by the lack of a common data format

\textsuperscript{3}Computational Biology Group, Department of Mathematics and Computer Science, University of Southern Denmark
Chapter 4. OpenLabNotes

Figure 4.1: The show note view in OpenLabNotes. A: The menu allows to select different types of lists or to create a new note. B: The history of the note, including checkmarks for valid signatures. C: Notes can be linked to samples, projects or notebooks. Moreover, files can be uploaded and attached and notes can be shared with other users. D: Here, the note is shown in a read only viewer. E: The note can be exported to PDF or DOCX files.

for ELN data storage and the fact that some products do not offer a backup mechanism in the first place. We therefore consider open source ELNs an attractive alternative to commercial products.

OpenLabNotes is an open source ELN extension for OLF that comes with advantages typically found in state-of-the-art note keeping tools, such as sharability, full text search, a reliable backup system, and a modern text editing experience. Using HTML for storing notes simplifies data export, long term storage, and the potential migration to another ELN. To our knowledge, OpenLabNotes is the first open source solution to provide a digital signature mechanism to prevent scientific fraud. Being open source, it can also be easily adapted to use alternative digital signature methods should the legal requirements change. The intended work-flow suggests that each note is signed by two authors, which serves to protect the researchers in a potential law case.
A common issue when introducing electronic data storage to a laboratory is that often not only an \textit{ELN}, but also an efficient sample management solution is required. Tightly integrating an \textit{ELN} with a \textit{LIMS} allows experimental information to be linked efficiently, thus increasing experimental documentation overall. While any open source \textit{ELN} can be integrated with some developmental effort into a \textit{LIMS} or vice versa, a single solution offers the advantage of reduced maintenance effort. If two independent systems are connected, an update of one system might lead to incompatibility with the other system. A fully integrated system such as \textit{OLF} is likely to include updates to all related functional parts at once.

In conclusion, \textit{OpenLabNotes} is a powerful and user-friendly alternative to costly commercial \textit{ELNs}. The digital signature mechanism implemented here follows accepted standards and should thus be suitable to fulfill legal requirements for data protection. Finally, the addition of \textit{OpenLabNotes} to \textit{OLF} effectively closes the gap between experimental documentation and sample management, making \textit{OLF} a comprehensive solution for functional genomics studies.

Despite its flexibility, \textit{OLF} is not ideally suited to deal with \textit{HTS} data, where samples are typically included in several plates belonging to the same screening library. A web application specifically accommodating this type of data is thus presented in the following chapter.
Chapter 5

SAVANAH: Sample Management and Visual Analysis of High-Throughput Screens

Objective of this Chapter
Development of a sample management system for high-throughput screening, covering
- management of molecular screening libraries
- tracking of diluted library copies and assay plates and corresponding readouts.
- user-friendly and interactive sample management.
- data export to suitable analysis software.

High-throughput screening (HTS) extends functional genomics and drug screening to large- or ultra-large scale (Chapter 1.6). The basis for this are molecular libraries, i.e. sets of plates containing the samples or reagents to be tested. A genome-wide siRNA screen, for instance, spans more than $200 \times 96$ well or more than $50 \times 384$ well plates. The elaborate process of generating and delivering screening libraries is addressed by a number of pharmaceutical companies such as Dharmacon™. The preparation, delivery, and storage of screening libraries, however, is costly. Libraries are therefore manufactured in a highly condensed form. Hypothetically, the solubilized reagents in a library plate can simply be resuspended, allowing for small amounts to be extracted to
produce a single set of assay plates. However, the quality of subsequent screens might be affected, since reagents will slowly degrade with each freeze / thaw cycle. Consequently, library plates need to be diluted in as few freeze / thaw cycles as possible. Assay plates obtained in this way can then be thawed and directly utilized for a screen when needed. However, fully diluting an entire library would lead to a very large number of plates that typically have to be stored at -20°C. This poses a storage problem, since freezer space is expensive and limited. To circumvent these problems, library plates are typically diluted in several steps (Figure 5.1). At first, each library plate is diluted and samples are divided between a number of master plates. These sets of plates, each representing the full library, are then stored in a freezer, except for one set of master plates, which is processed further. Again, the samples are diluted and transferred to a number of so-called mother plates. Only a single set of mother plates is in turn processed further and diluted into a number of daughter plates. Master, mother, and daughter plates are thus technically identical and only differ in the concentration of the sample, where only daughter plates are appropriately diluted for screening. Each set of daughter plates, also called assay plates, covers an entire screen. This process allows for a large portion of the screening library to be kept in a concentrated form, while reducing freeze / thaw cycles to a minimum. Depending on the actual concentrations, a screening library can thus be used for dozens of screens. A more comprehensive overview of the HTS procedure can be found in [123]. Over time, the thousands of samples found in a library are distributed across hundreds or thousands of plates. This demands a systematic solution for sample management.

![Diagram](image_url)

*Figure 5.1: HTS screening libraries are typically delivered in a highly concentrated form. To keep the number of freeze / thaw cycles low and to minimize the freezer space needed for the entire library, plates are diluted step-wise via master and mother plates, down to daughter plates, which can finally be used as assay plates in a screen.*
5.1 Requirements

A suitable system for supporting HTS needs to fulfill a number of requirements listed in the following as R1-28. These requirements are the basis for a comprehensive comparison with two existing applications in Supplemental Table A.4.

A sample management system for HTS should be open source (R1) to allow for adapting the system to the particular requirements of individual screening facilities. Moreover, access should be restricted based on different user roles (R2). The system should be able to handle the samples found in molecular screening libraries (R3). The multitude of plates representing different dilution stages of a library requires an efficient plate tracking system. Consequently, a HTS sample management system should model the entire library dilution process in a user-friendly way (R4). In addition, the system should keep track of empty wells in the library plates that are reserved for controls (R5). Occasionally, it might be necessary to allow the user to transfer samples to a different plate format, e.g. from 96 to 384 well plates (R6). Apart from monitoring the well volumes (R7), it would be advantageous to avoid considering inactive or deprecated samples. The system should thus allow for the corresponding well to be flagged as deprecated (R8) or even automatically assess the activity of all samples across different screens (R9). The sample information for all sets of diluted plates is identical, which suggests to avoid entering and keeping this information in the system redundantly (R10). Finally, only the fully diluted daughter plates are actually used as assay plates in a screen. For those plates, the sample information of each well needs to be linked to data originating from one or several experimental readouts (R11). Apart from information about an individual sample, experiment parameters need to be tracked that are often shared across a batch or an entire screen (R12). This includes, for instance, reagents that have been used or the cell lines that have been tested. Secondary screening can be guided by so-called cherry-picking (R13), where the most promising hits are transferred in silico to a new assay plate. Ideally, an automated mechanism for randomizing the sample position is included to avoid clustering similar reagents (R14). The system should cover typical use-cases like RNAi screens (R16), small compound screens (R17), as well as miRNA inhibitor and mimics screens (R18). All samples should be searchable (R19) and be linked to external databases (R20) to provide additional information to the user. Data should be visualized (R21-22) and subjected to quality control (R23). After appropriate normalization to counter known sources of variation (R24), hit discovery (R25) can be performed to filter for active samples that show an effect. In a typical screen, each sample is considered an independent experiment. There are, however, other types of HTS with related experiments, such as dose-reponse (R26) or time-series (R27) experiments, which would benefit from dedicated features. Finally, HTS readout data
can in theory be followed up in a secondary readout with RPPAs, suggesting to include support for linking such data (R28).

5.2 State of the Art

Most software systems that have been developed for HTS focus on the aspect of data processing and analysis and do not address the issue of sample management. Few LIMS have been developed in the past to accommodate small compound and RNAi screens. OpenBIS [124] is a general purpose LIMS that includes support for high-content screening (HCS), where the focus is mainly on image acquisition and processing. In HTS, however, only numeric readouts are produced. One system accommodating this type of data is Screensaver, which has been developed at the Harvard Medical School to manage siRNA as well as compound screens [125]. The system keeps track of screening libraries including the remaining volume of the wells. Reagents are linked to external databases to provide additional information. Furthermore, Screensaver provides cherry picking, which allows promising hits to be selected and to be assembled for secondary screening. Plates can be randomized and wells can be flagged as deprecated to exclude them from future screens. Normalization and hit detection are not supported and have to be performed externally. However, the results of such an analysis can subsequently be uploaded back into the system. An alternative to Screensaver is MScreen, which has been developed at the University of Michigan [126]. MScreen provides similar features and in addition supports dose-response curves, quality control, normalization, and hit detection. However, neither MScreen nor Screensaver support miRNA mimics or inhibitor libraries. In both systems, all sample information is uploaded via externally defined files. While this is an appropriate strategy to upload information about a screening library, which is only necessary once, it would be advantageous if users were offered a convenient and interactive interface to design smaller experiments manually. This motivated us to develop Sample Management and Visual Analysis of HTS (SAVANAH), an advanced sample management system for microtiter plates supporting small customized experiments as well as large-scale HTS. A comprehensive feature comparison between SAVANAH, Screensaver, and MScreen is provided in Supplemental Table A.4.

5.3 Results

SAVANAH aims at simplifying HTS data management for screening data, supporting both small experiments involving manually created plates, as well as HTS. To achieve the latter, SAVANAH offers a set of features for managing screening libraries efficiently.
5.3.1 Plate Layouts

In many cases, experiments involve several replicate plates to accommodate technical and biological variation. To avoid redundancy, SAVANAH thus separates the so-called plate layouts holding the sample information from the actual plates. This has the advantage that changes applied to the layout will automatically be propagated to the respective plates. To avoid long and cryptic sample names, sample information is divided in several categories shown in Figure 5.2. Each property is modeled as a distinct layer of the plate layout, which provides a clear data structure to the system. The user can conveniently change which sample property to edit, which will bring up the corresponding sample layer together with a box that allows the user to select a specific attribute. The user can apply attributes to individual wells or to multiple wells by drawing a box with the mouse. This mechanism enables users to quickly and efficiently design an experiment once the necessary sample properties have been entered into the system.

Figure 5.2: Plate layouts capture sample information in several layers and can be conveniently edited by the user.

5.3.2 Plates and Readout Import

In contrast to plate layouts, which hold sample information, plate objects identify physical plates by properties such as its plate type or its barcode. Moreover, plates can be
linked to measurements, where readout data can be uploaded for each plate individually or as a batch import. For the latter, the user is required to add all readout files to a zip file in which each file is named after the unique barcodes of the plate. *SAVANAH* accepts XLSX, comma, tab and semicolon separated files as input for readouts. During the import, the columns of the input file can be mapped to the expected properties, such as the well position and signal value. For convenience, this column mapping can also be stored as a so-called ResultFileConfig. The uploaded readout data can subsequently be visualized as a scatter plot or a heatmap (Figure 5.3).

![SAVANAH visualization of readout data. A: As a heatmap, where signal intensities are indicated by a color gradient in which red corresponds to a high signal and yellow to a low signal. B: As a scatter plot, where signal intensity is shown on the y-axis and the sample name on the x-axis. Only a few sample names can be shown to avoid overlays. However, both types of plots are interactive and provide additional information when hovering the mouse over a data point or tile.](image)

**Figure 5.3**: *SAVANAH* visualization of readout data. A: As a heatmap, where signal intensities are indicated by a color gradient in which red corresponds to a high signal and yellow to a low signal. B: As a scatter plot, where signal intensity is shown on the y-axis and the sample name on the x-axis. Only a few sample names can be shown to avoid overlays. However, both types of plots are interactive and provide additional information when hovering the mouse over a data point or tile.

### 5.3.3 Projects and Experiments

To keep a growing number of plate layouts organized, users are encouraged to group them into experiments, which in turn can be grouped into projects. First of all, this will make it easier to apply the search function efficiently. Moreover, a filter is available in the navigation menu. Here, a specific experiment and / or project can be selected by the user. Whenever the user opens a list view, e.g. a list of plate layouts or a list of plates, only items that are associated with the selected experiment and / or project will be displayed.

### 5.3.4 Screening Libraries

#### 5.3.4.1 Import Library

A screening library comprises a set of library plates with hundreds or thousands of molecules to be tested. For each of them, information about the plate and well location,
as well as sample and product identifiers needs to be added to the system. To this end, a tab separated file can be uploaded. SAVANAH expects this file to have a specific header (Supplemental Table A.5). SAVANAH will process this information and create a library together with the corresponding library plates. To avoid multiple database entries for the same sample, the system will check if an existing entry can be used by looking up the sample name and sample accession in the database. This functionality, together with the aforementioned function for importing readout data, have been implemented during an individual study activity by Martin Dissing-Hansen\(^1\).

### 5.3.4.2 Browse Library

After successful upload, the content of the library can be browsed conveniently in a tree structure (Figure 5.4). To find a specific entry of the library, the use of the search function in the navigation bar is recommended.

![Figure 5.4: Once uploaded to the system, libraries can be browsed in a tree view where wells are shown as children of plates. Clicking on a well provides information about sample it carries.](image)

### 5.3.4.3 Library Dilutions

Prior to use, library plates are serially diluted via master and mother plates to daughter plates, which are also called assay plates. Consequently, a large number of library plate copies need to be tracked. To avoid burdening the user with creating and linking these plates individually, SAVANAH introduces the concept of library dilutions. Each library dilution represents a full copy of the original library and can either be of type “Master”, “Mother”, or “Daughter”. In addition, diluted libraries of type “Mother” and “Daughter” keep track of their respective source. Diluted libraries are represented as a tree (Figure 5.5), where additional diluted libraries can be added at an arbitrary level. To this end, the user first selects an existing library or diluted library to be used as source. In addition to the number of additional diluted libraries to be created, the

\(^1\)Computational Biology Group, Department of Mathematics and Computer Science, University of Southern Denmark
Chapter 5. *SAVANAH*

system needs to be able to identify the corresponding physical plates unambiguously. To this end, proper labeling is imperative, which is typically achieved using barcodes. Ideally, the barcode contains information about the lineage of a plate, e.g. its index in the library, as well as indices identifying the diluted libraries the plates were derived from. To facilitate this during the creation of a set of diluted libraries, a barcode pattern needs to be specified. This pattern is required to include certain place-holders that are later substituted with the corresponding indices (Figure 5.5, right hand side). In the next step, the user can follow a step-by-step guide to use sets of daughter plates as replicates in a screen.

**Figure 5.5:** The dilution of screening libraries leads to a large number of plates that need to be tracked. *SAVANAH* uses the concept of diluted libraries to bundle all plates that belong to a single copy of a library. The lineage of each diluted library is shown in a tree view, where new copies can be added conveniently. In addition to the number of copies, the user is required to define a barcode pattern, which includes the library plate index and the indices of all diluted libraries that gave rise to a particular copy.

### 5.3.5 Create Experiment from Library

Experiments can either be created manually or from a screening library. For the latter, the user is guided through a number of selection steps (Figure 5.6). After selecting a library, the user can choose from all diluted libraries of type “Daughter” that have not been used in an experiment yet. Several diluted libraries can be selected at once to serve as replicates. Subsequent to selecting default properties, such as the cell line to be tested, the user is given the option to manually define a control sample for arbitrary wells. This is useful, since libraries typically have one or several empty rows or columns to be used for this purpose. Prior to assigning these control wells to the plates, the system will check if the corresponding well is empty. When the selection process is complete, the experiment will be created along with the following other objects (Figure 5.7). For
each library plate, a plate layout is created, which holds the sample information of the library and, in addition, the sample information entered by the user. Moreover, for each diluted library plate, a corresponding replicate plate is created. As mentioned before, replicate plates thus share the sample information. This allows users to edit the plate layout instead of editing each replicate plate individually. To equip the plates in this experiment with readout data, the user can now utilize the aforementioned batch import functionality.

5.3.6 Data Analysis

The data analysis, including normalization, quality control, and hit detection is currently not part of SAVANAH. Instead, the user can choose to export data directly to RNAiCe (Chapter 6), which is a dedicated R based web application suited for this task.

5.4 Discussion & Conclusion

A major challenge for any screening center that regularly performs HTS is the continuously growing number of plates and sample information that needs to be tracked efficiently. To make this elaborate task easier for the user, SAVANAH bundles diluted
Figure 5.7: Libraries that consist of library plates give rise to diluted libraries and diluted library plates. In an experiment, several diluted libraries can be selected as replicates for a screen. In this process, each library plate is first converted to a new plate layout holding the information of the library and in addition, experimental information, such as the cell line that was used. Moreover, for each replicate, a plate is created. These share the experimental information provided by the plate layout and can be linked to an arbitrary number of readouts.

Copies of library plates as abstract entities and handles the bulk of the plate setup automatically. This separates SAVANAH from similar tools such as Screensaver or MScreen. Another unique feature is the distinction between plate layouts and actual plates, which makes it possible to edit the sample information for all replicates at once. Finally, the interactive plate layout editor makes sample information accessible and editable in a convenient way. In addition to its support for HTS, SAVANAH is also ideally suited for designing and managing small customized experiments. This makes the system also attractive for laboratories that do not perform HTS, but deal with a large number of custom plate designs. The multi-level sample management may at first be considered as a burden, since users have to tediously add sample information before they can begin designing plate layouts and experiments. However, once this process is complete, sample properties can be applied quickly and efficiently, while the data model ensures consistency and efficient information retrieval. Finally, the possibility to export data to RNAice for further data processing and in depth analysis is a clear advantage compared to systems such as Screensaver that currently require manual data export.

In conclusion, SAVANAH provides plate based sample information for an entire laboratory in one place, which is a tremendous advantage over scattered spreadsheet files with cryptic and ambiguous sample identifiers. In contrast to comparable solutions, it streamlines the creation of library dilutions and the creation of screens with replicates. The powerful plate layout editor and the search functionality enable users to efficiently retrieve and edit all related sample information quickly. Additional plots and an in-depth analysis are facilitated through exporting data to RNAice, which is described in the following chapter.
Chapter 6

RNAice: High-Throughput RNAi and Compound Screen Evaluation for R

Objective of this Chapter

Development of an interactive analysis tool for high-throughput screening data, covering

- normalization of raw data to account for known sources of variation.
- quality control and data visualization.
- hit discovery and consensus hit lists based on several normalization strategies.
- determination of microRNA and drug targets.
- downstream functional enrichment analysis.

High-throughput screening (HTS) experiments produce a wealth of data that allows for functional analysis on the genome scale. The challenges in automation and miniaturization have been successfully addressed in the past, such that HTS is no longer exclusively used by pharmaceutical companies, but readily adopted by more and more academic institutions. HTS data analysis, however, is a challenging task complicated by the complexity of adapting cell culture conditions to the necessary experimental scale (Chapter 1.6). Individual plates or batches may suffer from quality issues, where transfection protocols are occasionally ineffective. In general, variation between batches,
plates and well positions complicate the robust identification of active samples. Spreadsheet tools, such as Microsoft Excel®, are still commonly used to tackle these challenges, but are unsatisfactory, since they are not suitable to deal with the large number of data points. Moreover, these tools offer only limited support for the advanced statistical methods required to normalize HTS data appropriately. Even if suitable macros were available, they would not provide the user with an efficient graphical user interface, which is better suited to guide the user through the different steps of the analysis.

### 6.1 Requirements

In the following, we define requirements an HTS analysis tool should fulfill. These requirements are denoted in more detail as R1-R39 in an exhaustive comparison found in Supplemental Table A.6.

#### 6.1.1 Data Input

Even though reporting guidelines for HTS experiments such as Minimal Information About an RNAi Experiment (MIARE)\(^1\) have emerged to standardize meta data, there is no such standard for HTS raw data. Even public databases such as PubChem\(^2\) or ChEMBL\(^3\) only provide minimal standardization. The common denominator is that raw data is typically reported as XLSX, comma, or tab separated file with interchangeable column names. At a minimum, such a file includes, for each well, an unambiguous identifier of the sample, a unique plate identifier, a well position, a replicate number (if replicates were included), and a signal value. To utilize this information, any analysis tool needs to include a column mapping step (R1) upon data import. Special care has to be taken of the way in which well positions are reported, which can either be alphanumeric (e.g. A01, A02, ..., B01, ...), numeric or with separate column and row indices (R2). Finally, it would be advantageous if screening data could be directly imported from public databases using identifiers such as the PubChem Assay ID (AID) (R3).

#### 6.1.2 Quality Control

Subsequent to the import of the data, a visual inspection should be possible (R4), including, for instance, plots for signal distribution, correlation of replicates and the

\(^1\)http://www.miare.org (access date: 06/01/2015)
\(^2\)https://pubchem.ncbi.nlm.nih.gov/ (access date: 06/01/2015)
\(^3\)https://www.ebi.ac.uk/chembl/ (access date: 06/01/2015)
variation between plates, batches, columns, and rows. Typically positive, negative, and transfection controls are incorporated on each plate to monitor the efficacy of cell transfection or treatment. Quality measures (R5-6) for the performance of positive and negative controls need to be included to allow experimenters to exclude plates that do not provide the necessary separation between active and inactive samples.

6.1.3 Normalization and Hit Discovery

Different normalization strategies should be provided to deal with the specific challenges presented by a particular dataset, including control-based (R7-8) and plate-based methods (R9-18). Moreover, the tool should offer various methods for hit discovery, including variance based methods (R19-20) and more sophisticated methods utilizing replicate and experiment-wide information (R21-22). The goal of hit discovery is to produce a list of active samples that qualify for further analysis both in vitro and in silico. Such a list should be exportable (R23) and visualization of the corresponding data (R24) should be offered. Here, interactive visualization (R25) is preferable over static plots, since this is better suited to present complex data without overwhelming the user. The analysis of several experiments or readouts in combination in so-called differential or multi-channel screening (R26) is advantageous to extract hits that show a specific effect. This could be, for example, the combination of a plus and a minus drug screen to exclude hits that generally cause toxicity. Since each normalization method has advantages and disadvantages, there is generally no unique method that can address all issues. Therefore, a consensus hit list (R27) should be offered, in which the hit lists of different normalization methods can be compared and aggregated.

6.1.4 Downstream Analysis

6.1.4.1 MicroRNAs

Optimally, additional resources are integrated to provide more information about the selected hits. This information can either be part of the tool or retrieved from public databases. In case of miRNA screens, additional information about mature miRNA identifiers, miRNA family membership (R28), disease-specific information, e.g. from miRcancer [127] (R29), as well as a mapping of miRNA hits to putative and experimentally validated target genes (R30) is advantageous.
### 6.1.4.2 Small Compounds

In case of a small compound or drug screen, a plot of the chemical structures (R31), a list of similarly structured compounds in the screen (R32), and a list of putative drug target genes would be desirable (R33).

### 6.1.4.3 Functional Enrichment Analysis

All screen types would benefit from a systematic functional enrichment analysis on the level of pre-defined gene sets that are associated with biological processes, functions, or pathways (R34-35). Finally, including de novo network enrichment (R36), in which potentially unknown pathways are extracted from existing interaction networks, such as BioGrid [128], is particularly suitable to generate new hypotheses and to identify the most promising hits in a screen.

### 6.1.5 User Interface

To make the analysis of HTS data easily accessible to users, a graphical or a web browser interface should be offered to guide to provide features for data input (R37) as well as for the presentation of results (R38). Moreover, it would be advantageous to implement the analysis following a reactive design, in which parameters of the analysis can be changed in the user interface without the need to reiterate the entire analysis process (R39).

### 6.2 State of the Art

A number of tools have been developed in recent years to deal with challenges typically encountered in analyzing HTS experiments. Even though commercial solutions like Spotfire® or Pipeline Pilot® support HTS data, we focus on non-commercial tools developed within the academic community. Here, analysis of genome-wide RNAi screens is supported by two different R Bioconductor packages. RNAither [129] and cellHTS2 [130] each provide features to process and normalize RNAi screening data. Both packages have the ability to produce HTML reports that allow the user to interact with the results in a user-friendly way. Moreover, a web service called web cellHTS2 has been established that provides convenient access to the analysis work-flow of cellHTS2 without requiring a local installation of R. Furthermore, both packages can link identified genes to gene ontology terms [131] and test for gene set enrichment [132]. While both tools support gene ontology analysis, only cellHTS2 results can be subjected to more sophisticated
functional enrichment analysis via the HTSanalyzeR package [133]. Another noteworthy tool is HiTSEE KNIME [134], which focuses on the analysis of small compound screens. While it lacks many common features in data normalization and hit discovery, it offers a powerful user interface to explore small compound data, including structural comparisons and the ability to explore the so-called neighborhood of a compound, which consists of structurally similar compounds.

In summary, RNAither and cellHTS2 provide excellent support for RNAi screening experiments, but fail to accommodate alternative screen types such as small compound or miRNA screens. Moreover, they lack hit discovery methods that optimally utilize sample replicates, such as the Bayesian hit detection method suggested by Zhang et al. [135]. The strict separation between input parameters and results does not allow the user to explore different thresholds and data processing strategies interactively and without manually repeating large parts of the analysis procedure. We envision that a more responsive web application would motivate users to explore and compare alternative data processing strategies and thus to better accommodate the unique challenges that come with each individual experiment. This motivated us to develop RNAi and Compound Screen Evaluation (RNAice), which in addition to RNAi and CRISPR/Cas9 screens also offers support for small compound and miRNA screens with their unique challenges in downstream analysis. A comprehensive comparison between cellHTS2, RNAither, and RNAice can be found in Supplemental Table A.6.

6.3 Methods

Processing HTS data involves a number of steps. Raw signal data needs to be normalized to account for known sources of variation. The quality of the screening data needs to be assessed by visual inspection of the signal distribution and by measuring the separability of included controls. The major goal of primary HTS is to obtain a list of hits, i.e. samples that show a significant change in phenotype. Various methods are available to make this distinction. A detailed overview can be found in [136] and [55]. Methods implemented in RNAice are briefly introduced in the following.

6.3.1 Normalization of Raw Signal

While some hits are easily spotted both visually and statistically due to their extreme effect, there are typically many hits with moderate yet significant effect. The specificity and sensitivity for identifying these hits depends largely on the ability to remove noise from the raw data. In general, two types of normalization methods exist, namely control
based and plate based normalization.

6.3.1.1 Control Based Normalization

Control based normalization methods rely on a series of wells containing negative control samples that demonstrate little or no effect and positive control samples that exhibit a strong effect. Control samples can be used as a reference point to measure the relative effect observed in samples located on the same plate as the controls. This allows for inter-plate comparisons even if absolute values vary significantly between plates and batches, since the control normalized values express the signal of the various plates as a percentage of effect strength. Two control based normalization methods are presented in the following.

**Percentage of control (POC):** Here, only a single control type is needed to establish a reference point, i.e.

$$POC = \frac{x_i}{\bar{c}} \cdot 100$$  \hspace{1cm} (6.1)

where $\bar{c}$ corresponds to the control mean and $x_i$ corresponds to the sample value.

**Normalized percentage inhibition (NPI):** Here, two controls establish an effect range that can be computed as a percentage, i.e.

$$NPI = \frac{\bar{c}_+ - x_i}{\bar{c}_+ - \bar{c}_-} \cdot 100$$  \hspace{1cm} (6.2)

where $\bar{c}_-$ corresponds to the mean of the negative control and $\bar{c}_+$ corresponds to the mean of the positive control, respectively.

6.3.1.2 Plate Based Normalization

Control based normalization is often considered inappropriate, since the number of control wells is typically kept small to reserve more space for samples. Moreover, control wells are typically located on the outer wells, where the signal is often strongly biased by evaporation. Most importantly, control samples may occasionally not perform as expected, e.g. in cell viability assays, where negative controls may prove lethal for a particular cell line or where a cell line may be resistant towards a normally lethal positive control [137]. A viable alternative is plate based normalization, which operates
under the assumption that the majority of reagents do not demonstrate any significant effect. Consequently, most wells in a plate qualify as negative controls, which largely increases the number of wells contributing to a virtual reference point. A control reagent may show unexpected behavior in some scenarios, while the plate-based normalization is more robust due to the large number of contributing samples [136]. Plate based normalization, however, is limited to primary screening. Follow-up screens are typically confirmation screens of hits found in the primary screen, where this assumption does not hold. Consequently, control wells are typically included in primary screening even when only plate based normalization methods are considered. The control wells will then be used at a later stage to compare the effect strength between primary and secondary screening results. Finally, screening libraries are sometimes not randomized but clustered. In this case, it is quite likely that groups of wells on a plate demonstrate comparable effects. Examples are small compound libraries, where structurally similar compounds are often grouped or siRNA libraries, where functionally related reagents may be located on the same plate. The most frequently used plate based normalization methods are presented in the following.

(Robust) z-score: The z-score expresses effect strength as a function of the general variability of the data, i.e.

$$z - \text{score} = \frac{x_i - \bar{x}}{SD_p}$$

6.3

where $x_i$ corresponds to the sample value, $\bar{x}$ corresponds to the plate mean and $SD_p$ corresponds to the standard deviation (SD) of plate $p$. A disadvantage of this method is that it considers all samples for computing mean and SD. Alternatively, the robust z-score can be used, where mean and SD are exchanged with median and median absolute deviation (MAD) respectively. In z-score normalized data, the plate mean is 0 and the SD is 1. This type of normalization corrects for general differences in signal intensity and expresses effect strength in dependence of the general signal variation, thus allowing for inter-plate comparison.

B-score: A common problem in HTS are positional effects. These can be caused by, e.g. increased evaporation of the outer wells or by a technical bias introduced in cell seeding, where each row is typically supplied by a different tube. The result of these effects is a signal bias that is in most cases row and column specific. To mitigate this, Brideau et al. suggest the B-score [138], which utilizes Tukey’s two way median polish to obtain a signal estimate $r_{ijp}$ that is corrected for position specific bias $\hat{x}_{jip}$. It includes the estimated average of the plate $p$ as $\hat{x}_p$, the estimated offset of row $i$ in plate $p$ as $\hat{R}_{ip}$
and the estimated offset of column $j$ in plate $p$ as $\hat{C}_{jp}$:

$$
\begin{align*}
    r_{ijp} &= x_{ijp} - \hat{x}_{ijp} \\
    &= x_{ijp} - (\bar{x}_p + \hat{R}_{ip} + \hat{C}_{jp})
\end{align*}
$$

Similar to the robust z-score, the B-score can be obtained by dividing the corrected signal estimate by the MAD of the plate:

$$
B - \text{score} = \frac{r_{ijp}}{\text{MAD}_p}
$$

The B-score is ideally suited to deal with positional effects, but particularly in 96 well plates it may introduce an additional bias in case of rows or columns that contain many active samples.

### 6.3.2 Quality Control

Quality control (QC) in HTS is imperative to identify batches or individual plates that did not perform as expected, e.g. if the transfection was inefficient or if the cell viability was affected. Moreover, QC helps to monitor time dependent effects where the quality typically decreases with each following plate, thus limiting the maximal batch size. Several parameters of QC can be investigated visually, such as row and column means to check for positional effects, or scatter plots to investigate the signal distribution. Ideally, wells with positive and negative controls are available on each plate. Two established measures that assess how well positive controls can be separated from negative controls are presented in the following.

#### 6.3.2.1 Strictly Standardized Mean Difference

Acknowledging that QC measures like the signal-to-noise and signal-to-background ratio fail to capture the variability of the data appropriately, Zhang et al. suggested a dimensionless parameter called Z-factor [139]. It is defined as the ratio of the difference between sample and control mean, and the dynamic range of the signal. This measure, called Z'-factor when comparing control values, is a widely used QC measure in HTS. However, Zhang et al. have criticized it for a lack of solid statistical interpretation and
proposed an alternative score called strictly standardized mean difference (SSMD) [140], which is defined as:

$$SSMD = \frac{\mu_1 - \mu_2}{\sqrt{\sigma_1^2 + \sigma_2^2}}$$  \hspace{1cm} (6.7)

where $\mu_1$ and $\mu_2$ are the means and $\sigma_1$ and $\sigma_2$ the SDs of two populations 1 and 2\(^4\). Like the Z-factor, the SSMD captures not only the difference, but also the variability of both populations. In contrast to the Z-factor, however, the SSMD is easier to interpret. An $SSMD > 3$ indicates that the mean difference is at least three times the SD of the difference of the two populations. Moreover, an $SSMD > 3$ indicates that the probability that a value from the first population is larger than a value of the second population is close to 1 (0.99865), adding a probability interpretation. Following the three sigma-rule for significance, a $SSMD$ of 3 is thus a suitable cutoff for a pass / fail test in QC.

### 6.3.3 Hit Detection

The goal of HTS is to identify active samples or reagents in a screen that show an effect. The threshold at which an observed effect is considered significant depends on the variability of the data. Therefore, a common and straightforward approach to detect hits is to define a threshold based on $k$ SDs or, to increase robustness, on $k$ MADs. The factor $k$ is chosen by the user to control the number of hits to be selected. A small $k$ results in less stringent filtering and is likely to include many false positives, while a large $k$ will lead to stringent filtering with and lead to many false negatives. In reality, the choice of $k$ mostly depends on economic considerations, where as many promising hits as possible are included in a secondary confirmation screen. Therefore, the false discovery rate (FDR) and its control via multiple testing correction are widely ignored in primary screening.

#### 6.3.3.1 SSMD for Hit Detection

The intuitive approaches of defining a window based on SD or MAD are often criticized for being relative arbitrary. Moreover, these approaches are not suited to fully utilize replicates. An intuitive alternative that is often considered for HTS experiments with replicates is thus the t-test, which can be used to assign a p-value to the difference.

\(^4\)Note: populations here can refer to the comparison of two controls or a sample and a control.
between sample and control replicates. However, Zhang et al. demonstrated the t-test is in fact an inappropriate measure of effect strength, due to its dependence on the number of available replicates [141]. A suitable alternative for hit detection that separates the effect size of the impact of the number of replicate samples, is the SSMD score described in Chapter 6.3.2.1.

6.3.3.2 Bayesian Hit Detection

The hit discovery methods presented so far can be used to assess and rank the activity of all samples in a screen. A common disadvantage is that the FDR is unknown but expected to be inflated, since a large number of samples are tested independently. Moreover, variation between plates and batches is only taken into account on a per plate basis. While this strategy is generally appropriate, it might lead to misleading results if individual plates contain clusters of active samples. Zhang et al. therefore propose an alternative hit discovery based on a Bayesian model [135]. One of the main advantages of this method is that it calculates effect strength per plate while borrowing information from the entire experiment. Moreover, this model maintains a balance between contributions of sample wells and control wells. In practice, the experiment-wide information is used to calculate the priors for the model, while the actual likelihood is calculated per plate. Another major advantage of this method is that it allows one to effectively control the FDR via multiple testing correction, e.g. using the method of Benjamini-Hochberg [142]. Bayesian hit detection thus offers a statistically motivated alternative for defining a threshold for hit detection. A detailed description of the method, which was implemented for RNAice by Martin Dissing-Hansen\(^5\), is included in Chapter D.1.

The method presented here is more robust to plate-specific signal bias due to its ability to utilize information of the entire experiment or batch, yet it does not account for positional effects. Similar to control based normalization methods, its effectiveness depends crucially on the reliability of controls. Pooling the variance of negative controls across plates, however, allows for this method to be used even if relatively few negative control wells are included. This is of particular importance for 96 well plates, where typically only very few controls are available. The result of hit discovery in RNAi or CRISPR/Cas9 screens is a list of genes that are associated with a particular phenotype. In contrast, miRNA inhibitor or mimics screens require a list of target genes.

\(^5\)Computational Biology Group, Department of Mathematics and Computer Science, University of Southern Denmark
6.3.4 MicroRNA target genes

MiRNAs operate as key regulators of biological processes by inhibiting the translation of up to hundreds of genes (Chapter 1.5.1.2). Consequently, the effect of inhibiting or mimicking a miRNA can only be understood through interrogating the function of the affected genes. It is therefore imperative to acquire knowledge about miRNA-gene interactions prior to any functional analysis. A number of prediction algorithms have been developed to solve this problem computationally. Common features considered are the requirement for a perfect match in the seed region, evolutionary conservation of the target site sequence, as well as the free binding energy at and the accessibility of a particular genomic site. More fine-grained strategies exist and have been reviewed exhaustively in [143, 144]. The predictions of these tools are available through the Bioconductor package RmiR, including predicted targets from mirbase [145], targetScan [146], miRanda [147], miRDB [148], and PicTar [149], as well as experimentally validated targets from tarbase [150]. In addition, a database of human miRNA targets has been created using the tool RNAhybrid [151]. RNAhybrid predicts miRNA target interactions via the free binding energy. The advantage of this database, named RNAhybrid_hsa, is that each miRNA target interaction is associated with a p-value. This allows for controlling the sensitivity of the target prediction. A general weakness of miRNA target prediction is the high false positive rate, where a single miRNA is typically predicted to target hundreds of genes. Controlling this parameter in RNAhybrid_hsa is thus advantageous.

Ideally, predicted miRNA target interactions are verified experimentally. Reliable experimental evidence for miRNA target interactions is available through reporter assays [152]. However, this requires experimenters to tediously validate each single interaction. Recently, more systematic approaches such as crosslinking immunoprecipitation (CLIP) have become available [153]. Here, the miRNA and its target mRNAs are crosslinked in the RISC complex, which is subsequently extracted via immunoprecipitation. In CLIP-seq, the degradation of the RISC complex is followed by NGS, which allows for the identification and confirmation of miRNA targets. However, many potential interactions are likely to be missed, since they can only be detected if both the miRNA and the corresponding target gene are actively transcribed in abundance [154]. Therefore, prediction algorithms currently remain the primary source of miRNA target information. To accommodate the high false positive rate of these methods, we propose a filter for high-confidence target genes introduced in the following.

6.3.4.1 High-Confidence miRNA Target Genes

In a typical miRNA experiment, putative target genes are aggregated for all miRNAs hits using one of the established public databases or prediction methods. The resulting gene list is then subjected to further analysis (Chapter 6.3.5). However, several studies demonstrated that functional miRNA target enrichment analysis leads to erroneous results [155, 156]. Godard et al., for instance, found that pathways are reported as enriched independent of the disease investigated in the study [155]. This motivated the authors to investigate and compare lists of randomly selected miRNAs, confirming the suspicion that miRNA target enrichment is unspecific. We therefore propose to filter the miRNA hit list to exclude uninformative miRNAs. To this end, we utilize the hypergeometric test to compute the probability of drawing \( k \) successes from a population of size \( N \) in \( n \) draws, given that there are \( K \) success states in the population:

\[
P(X = k) = \binom{K}{k} \frac{(N-K)}{\binom{N}{n}}
\]  

(6.8)

This allows us to calculate the probability that \( k \) out of \( n \) miRNAs target a particular gene given that this gene is predicted as a target for \( K \) out of \( N \) total miRNAs. Since we are in fact interested in the probability of observing at least \( k \) miRNAs that target a particular gene, we can calculate the cumulative probability as:

\[
P(X \geq k) = \sum_{i=k}^{n} \binom{K}{i} \frac{(N-K)}{\binom{N}{n}}
\]  

(6.9)

Genes with a significant p-value are thus found more frequently as targets in the miRNA hit list than we would expect by chance. To compute these p-values, we first needed to know \( K \) for each of the genes. However, for RNAhybrid_hsa, the value of \( K \) depends on the p-value threshold selected for miRNA target interactions in the first place. We thus added additional tables to RNAhybrid_hsa that yield corresponding \( K \) values for p-value thresholds of 0.05, 0.01, 0.001, and 0.0001.

To apply this strategy irrespective of the miRNA target database used, we implemented a second method based on repeated sampling of miRNA target lists. These lists of equal length as the original hit list are utilized to extract the putative target genes. In this way, the average number of miRNAs targeting a particular gene can be computed to obtain an estimate of \( K \).
Finally, the resulting p-values from the hypergeometric test are corrected for multiple testing using the Benjamini-Hochberg method [142]. The remaining target genes that pass the significance threshold after this correction are considered as high-confidence target genes for further analysis.

### 6.3.5 Analysis of Candidate Genes

Both, siRNA and CRISPR/Cas9 screens yield hit lists in the form of genes. Similarly, the effect of miRNAs and small compounds can be best understood under consideration of their target genes. It is thus crucial to extend the HTS analysis to the systematic functional analysis of gene sets. Three popular approaches that have been included in RNAice are briefly discussed in the following.

#### 6.3.5.1 Gene Set Overrepresentation Analysis

Several databases exist that link sets of genes to functional categories in gene set over-representation analysis (GSOA). One of the most popular providers of gene sets is Gene Ontology [131], which groups genes across three categories, namely cellular compartment, molecular function, and biological process. Being organized as an ontology, these so-called GO terms form a hierarchical structure, in which more generalized terms, which are shared by many genes, are on top, while more specific terms, which are shared by few genes, are found on the bottom. Other popular databases that link gene sets to functions are KEGG [157] and Reactome [158], which consider sets of genes as biological pathways that provide a specific function to the cell or that are related to a specific disease. Utilizing the hypergeometric test, a probability can be calculated for observing a given overlap between the genes in the hit list and pre-defined functional gene sets [159]. Overrepresentation analysis identifies gene sets for which significantly more genes are found than would be expected by chance.

#### 6.3.5.2 Gene Set Enrichment Analysis

In GSOA analysis, only the membership of a hit gene in a gene set is considered. In contrast, gene set enrichment analysis (GSEA) takes the rank of all genes of a set into account [132]. To this end, an enrichment score (ES) is calculated as a running sum for each gene set S. The ES is increased if a gene is in S and decreased otherwise. The magnitude of the increase or decrease depends on the correlation of the gene with the phenotype, i.e., in case of HTS it is the measured effect strength. By permuting the phenotype labels repeatedly, a null distribution of the ES can be calculated. This allows
for the computation of significance values for each individual score. In this way, the ES indicates whether members of a gene set are enriched at either end of the ranked sample list. GSEA, which is based on the Kolmogorov-Smirnov statistic [160], is generally considered more powerful than the hypergeometric test described above, but is often criticized for a lack of sensitivity [161].

6.3.5.3 Network Enrichment Analysis

Tamayo et al. could show empirically that the underlying assumption of GSEA, the independence of gene effects, does not hold. This indicates a severe bias of the ES due to frequent gene-gene interactions [162]. Moreover, the results of GSEA as well as GSOA depend crucially on the selection of the a priori defined gene sets [163]. Both of these issues can be mitigated by network enrichment analysis, which utilizes biological networks that consist of experimentally validated or predicted interactions between genes and/or proteins [128, 164, 165]. In contrast to methods based on gene sets, network enrichment allows for functional pathways or functionally related gene sets to be discovered de novo. Analyses based on pre-defined gene sets or pathways, such as those found in KEGG or Reactome, are subjected to a research bias, whereas de novo network enrichment relies only on knowledge of general interactions between genes or proteins and thus allows for extracting pathways without this bias. To utilize such methods, gene regulation or protein-protein interaction networks such as BioGrid are needed. While the resulting pathways often overlap with the pre-defined ones found in KEGG, for instance, there are also pathways that have not been formally described and can potentially identify unknown mechanisms. These can be further investigated and might lead to an explanation of the observed phenotype. Several distinct algorithms and tools are available to achieve network enrichment [166–168].

We chose to include KeyPathwayMiner (KPM) [166], which features two enrichment models to extract de novo pathways. KPM operates on a binary indicator matrix of cases x entities, where cases correspond to samples or patients and entities to genes or more generally to the nodes of the network. The common goal of both models is to find connected components that are enriched for entities that are active in “most” cases. Individual Node Exceptions (INES) is an enrichment model that introduces two parameters K and L to control what “most” cases means. K defines entity exceptions, i.e. in each sub-graph up to K nodes can be inactive. Moreover, L defines case exceptions, meaning that an entity is counted as active if it is active in all cases but L. A consequence of the K parameter is that KPM often reports sub-graphs that are highly similar and only differ in the up to K nodes that have been added arbitrarily at the periphery. These so-called border exception nodes can be removed by KPM to avoid
this issue. One disadvantage of INES is that it tends to use the flexibility introduced by the $K$ parameter to add well connected hub nodes to the sub-graph. Although these are selected to increase the overall size of the solution, this effect might not always be desirable. Therefore, an alternative enrichment model called Global Node Exceptions (GLONE) omits the $K$ parameter in favor of a globally distributed number of case exceptions defined by $L$. In other words, all entities in a sub-graph are active in all cases but $L$. $KPM$ was selected, since, in contrast to other methods, the parameters are easily interpretable. Moreover, $KPM$ offers a web service that can be integrated to allow the user to save time by performing the demanding computations for de novo network enrichment asynchronously on a powerful server. In this way, the user can conveniently continue to explore the HTS results while waiting for the results. The sub-graphs revealed by de novo network enrichment are likely to provide new insights into molecular mechanisms that were affected during the screen and which should be followed up in further experiments (see Chapter 6.4.2.1 for an example).

6.4 Results

In the following, we describe the user interface and typical work-flow in RNAice, including options for data input, data exploration, quality control, normalization, hit discovery, and functional analysis. This chapter concludes with two application cases that demonstrate the capabilities of RNAice on previously published RNAi and miRNA mimics screening data.

6.4.1 User Interface

6.4.1.1 Input

RNAice is flexible with regards to the input format (Figure 6.1). The start page of the application is the “Input” tab, where the user can choose between several options. To get familiar with RNAice, the user can select one of of several published data sets, including a miRNA mimics screen of cell viability in human melanoma cells [169], a differential genome-wide siRNA screen aimed at identifying modulators of tumor necrosis factor $\alpha$ (TNF$\alpha$) [170], and a genome-wide siRNA aimed at identifying genes implicated in the sensitivity of human colon cancer cells to the anti-cancer drug vorinostat. The latter is complemented by a genome-wide miRNA mimics screen [171]. Alternatively, users can provide an AID or upload their own dataset. The raw data is shown in a data table at the bottom of the page. Data tables shown in RNAice are interactive, i.e. the content can be sorted by column in ascending or descending order. Moreover, the content can
be searched with a global search box located in the top right corner of the data table or with column-specific search boxes located at the bottom of each column. Entering a search term will filter for table rows that match the query.

Subsequent to uploading or selecting a dataset, the respective columns need to be mapped to properties known to and required by RNAice. This can be achieved by clicking on a checkbox called “Show file input options”, which provides select boxes for identifying properties, such as plate id, well location, accession number, measured signal, replicate, experiment and control status of each row, by the column name of the input data. Moreover, the type of screen, type of accession number, type of well position as well as specific positive and negative controls need to be selected. The user can decide to log-transform the data and to include the calculation of B-scores, which prolongs the run-time considerably due to the complexity of the method. A click on the button “Process raw data” triggers pre-processing and normalization of the data. Upon completion, additional options become available in the top menu.

**Figure 6.1:** A: RNAice guides the user through the analysis of an HTS dataset. To this end, the functionality is grouped into categories that are accessible through a navigation menu on top. At first, only the “Input” tab is available, while additional functionality becomes available after proper processing of the data. B: As input data, users can either select one of the included demo data sets, import a dataset by its PubChem id or upload their own data file. C: Since the composition and properties of any uploaded files is unknown, a form is provided to enable the user to identify the required columns and to enter additional information, e.g. about the type of the screen or the type of accession id provided.
6.4.1.2 Data

The “Data” tab provides a series of plots to explore the processed data (Figure 6.2). A table shows the pre-processed data, which now includes columns for (robust) z-score, for the signal centered by mean or median, and optionally for the B-score and for control normalized values. Scatter plots show the entire data of the screen and are divided by experiment and readout. Plates and replicates are distinguished by color and shape, respectively. In the plots included here, the raw signal can be compared to different normalization methods enabling users to perform a direct comparison of their effect. Density plots provide an impression of the distribution of the raw data across replicates, experiments and readouts. A quantile-quantile plot is available to assess if the signal data is normally distributed. Finally, an interactive plate viewer allows users to select individual replicate plates for heatmap and scatter plots. The quality of the screen can be further investigated in the following tab.

6.4.1.3 Quality Control

The “Quality Control” tab depicts the variability of the screen and the performance of the controls (Figure 6.3). A plot of signal spread across the entire screen consists of box plots of each plate to demonstrate in how far the quartiles of different plates agree. The replicate correlation plot includes linear regression models and $R^2$ values for all pairwise combinations of replicates in the screen. A plot dedicated to the spread of control values consists of a box plot for each control and plate. In this way, problematic controls and/or plates can be spotted. The separability of controls is an important indicator for the quality of a screen, since it indicates the expected sensitivity for hit discovery. As described earlier (Chapter 6.3.2), the $Z'$-factor and SSMD are both suitable methods to quantify separability of samples or controls. Since the SSMD can be considered superior to the $Z'$-factor, it is currently the only method included in RNAice. The corresponding plot depicts the SSMD of all pairs of positive and negative controls across all plates. Finally, the column and row means are plotted for each replicate plate after centering the raw signal data with the mean. This plot is particularly suited to spot positional effects. Here, a characteristic u-shape indicates a signal increase of the outer wells due to, for instance, increased evaporation. After successful assessment of the quality of the screen and sources of variation, the user can proceed to hit discovery.
Figure 6.2: RNAice allows its users to explore screening data by plots such as A: Scatter plot of samples in the screen ordered by plate (color), replicate (shape) and well number. B: Signal density plot. C: quantile-quantile plot. D: Interactive heatmap of a plate, where the color gradient corresponds to signal strength. E: Interactive scatter plot of a plate with the well index on the x-axis and the signal strength on the y-axis. Error bars indicate the standard deviation of the replicates.
Figure 6.3: RNAice produces a series of plots for quality control. A: Box plots of all plates and replicates. B: Replicate correlation plot including a linear regression (black line) with corresponding $R^2$ correlation factor. The grey line indicates the identity. C: Signal spread of the various controls across all plates. The positive controls rela and trnf1 are more robust than the negative controls sicon and lrp5. D: Control separability measured by SSMD. Here the separability between the negative control lrp5 and the other controls is shown. The red area indicates bad separability and the orange area good separability. An SSMD outside of the orange area indicates excellent separability. E: Plate signal is centered by the mean and subsequently column means are plotted across all plates and replicates to indicate positional effects typically evident by a u-shape.
6.4.1.4 Hit Discovery

The “Hit Discovery” tab is dominated by an options pane, which allows the user to include one or several experiments and readouts. Moreover, a normalization method and hit detection method need to be selected together with a sensible margin that is used to compute thresholds for active samples (Figure 6.4). In case of the Bayes method, a p-value threshold has to be selected instead. Finally, the user can select whether to show all hits, or to filter for promoters or suppressors. Sample filter options are provided to exclude or include specific samples from the final list. If several experiments and/or readouts are selected, differential screening can be selected to use one pair of experiment and readout as a baseline or reference. Hits discovered in this reference will be removed from other selected experiments or readouts.

In response to the settings selected above, a list of hits is generated. Depending on the screen type and the type of accession number, additional information is added, such as, for instance, entrez gene ids, gene symbols, miRNA ids, or miRNA family ids. An alternative view of the hits is provided in an interactive scatter plot depicting the hits of the screen in decreasing order. For a large number of samples, the labels on the x-axis quickly overlay and thus become unreadable. The user can, however, still move the mouse cursor over one of the hits to trigger a tooltip with additional sample information. Finally, a large heatmap of the entire screen is plotted, where hits are indicated through small black arrows (Supplemental Figure B.6). This form of presentation allows to quickly assess systematic problems in the hit discovery, e.g. if hits accumulate on a few plates or if they show a tendency to occur in a specific row or column. In this way, the user is encouraged to critically review the selected normalization and hit discovery strategy. However, often none of the available methods seem to be fully convincing, since they all have their particular advantages and disadvantages. The next tab is therefore dedicated to comparing multiple methods in simultaneously.

Consensus Hits:

The “Consensus Hits” tab offers to consider several normalization methods in parallel (Figure 6.5). A consensus hit list is generated based on all selected methods. The user can define a minimal number of methods that need to report a particular hit for it to be included. Similar to the ordinary hit list, the consensus hit list can be viewed as a table or as a scatter plot. In addition, a Venn diagram shows to what extent the different methods agree and how many of the hits are exclusive. The remaining tabs in RNAice depend on the screen type and are described in the following.

MicroRNAs: The “MicroRNAs” tab can only be selected after successful identification of hits in a miRNA screen. Here, additional information is provided for further analyses.
Figure 6.4: RNAice provides several options for hit discovery. A: The user can choose to include or exclude samples based on a regular expression. Moreover, differential screening is possible, where hits identified in one experiment or readout are removed from others. B: The user selects which experiments and readouts to consider. C: Several normalization and hit detection methods are available. A threshold needs to be defined and optionally a direction of the effect can be selected. D: The result is shown as either a data table (Hit List), as an interactive plot (Hits Plot), or as a heatmap that encompasses the entire screen.

Figure 6.5: The problem of experimental variation in the data can be accommodated by normalization, where each method disqualifies some hits while promoting others. A: This is reflected in consensus hit list where several methods can be selected in parallel. A threshold can be adjusted to include only hits reported by several methods. B: The consensus hit list can be viewed as a data table or interactive plot. C: A venn diagram illustrates to what extent the methods agree.
A table of predicted or experimentally validated miRNA target genes can be generated using a number of different databases (Chapter 6.3.4). Moreover, a miRNA family view provides an overview of family coverage, i.e. how many members of a miRNA families appear as hits in this particular screen. Cutoffs can be selected for family size and percentage of coverage to filter for interesting families. The idea behind this is that miRNAs in a family share the same seed sequence and thus a large number of target genes. Single miRNA hits may be due to an off target effect. In contrast, observing several miRNA mimics or inhibitors in the hit list that are part of the same miRNA family is unlikely to be the result of off-target effects.

The effect of miRNAs is attributed to their target genes. RNAice enables users to select predicted or experimentally validated target genes from a selection of sources. As discussed in Chapter 6.3.4.1, the list of miRNA targets is likely to contain a large number of false positives. To mitigate this, high confidence target genes can be computed. Finally, the integration of the miRcancer database [127] points the user to literature of known associations of the miRNA hits with various types of cancer.

Small Compounds:

Small compound screening aims at identifying drugs that are suitable to influence cellular function through so called drug targets, i.e. proteins that interact with a chemical compound. Several million drug target interactions have already been collected in databases such as STITCH [172]. RNAice uses STITCH to identify possible drug targets of active compounds found in small compound screens. Subsequent functional analysis on the resulting gene list is suited to reveal if several of these drugs achieve an effect through the same or associated proteins or molecular functions.

Systematic Functional Analysis on the Gene Level: The systematic functional analysis of gene lists through GSOA and GSEA (Chapters 6.3.5.1 and 6.3.5.2) is facilitated through the integration of the R package HTSanalyzeR [133]. In the case of siRNA or CRISPR/Cas9 screens, either the hit list or the consensus hit list has to be selected. Alternatively, if a miRNA is analyzed, the choice is between the original miRNA target list or the high confidence miRNA targets. Finally, if a small compound screen is selected, the drug target gene list is pre-selected. The user can choose several gene sets to be considered, including the three categories of Gene Ontology and KEGG pathways. Moreover, the p-value cutoff, the minimal gene set size, the number of permutations for GSEA and the selected method for multiple testing correction can be adjusted. The “Start analysis” button will trigger an analysis. Alternatively, de novo network enrichment analysis (Chapter 6.3.5.3) is available through integration with the web service of...
the *KPM* web application\textsuperscript{7}. The user can adjust the parameters of *KPM* and initialize the analysis remotely by pressing “Start KPM”. During the analysis, the user can freely explore other results provided by RNAice. Upon completion, the key pathways, i.e. gene sets, provided by *KPM* will be shown in a graph. In contrast to the result graph shown in the *KPM* web application, RNAice will include additional nodes and edges depicting the *miRNAs* that share the identified genes as targets.

### 6.4.2 Application Cases

In the following, we demonstrate how RNAice can be utilized to analyze HTS raw data. The first example is a genome-wide siRNA screen aimed at identifying modulators of tumor necrosis factor alpha (TNF\(\alpha\)) \cite{170}, which is implicated in inflammatory diseases and cancer. Our aim here is to demonstrate the basic data processing and quality control capabilities of RNAice and to compare our results to the results reported in the study. We also apply functional enrichment analysis to evaluate the resulting hit list in a wider biological context. Another example is a synthetic lethal screen in colon cancer cells aimed at detecting resistance genes for vorinostat, a drug for cancer treatment \cite{171}. In addition to a genome-wide siRNA screen, Falkenberg et al. performed a *miRNA* mimics screen. The combination of these two screening types in a single experiment provides an opportunity to test the *miRNA* high-confidence target method introduced in Chapter 6.3.4.1.

#### 6.4.2.1 A Genome-Wide RNA Interference Screen Identifies Caspase 4 as a Factor Required for Tumor Necrosis Factor Alpha Signaling

Nickles et al. have performed a genome-wide differential *siRNA* screen for genes modulating TNF-\(\alpha\), which also affects NF\(\kappa\)B signaling \cite{170}. The screen includes 56 duplicated 384 well plates. In order to distinguish TNF\(\alpha\) modulators from genes that affect cell viability in general, two readouts were performed. A *Renilla* luciferase acting under control of a \(\beta\)-actin promotor serves as a general reporter of cell viability (channel 1). In addition, a second reporter plasmid contains eight NF\(\kappa\)B binding sites that control the expression of firefly luciferase (channel 2). Since the NF\(\kappa\)B pathway is triggered by TNF\(\alpha\), this allows for the identification of potential modulators. The screen includes the two negative controls sicon (siCON) and lrp5 (siLRP5) as well as two positive controls rela (siRELA) and tnf1 (siTNFR1). Inspecting the raw signal data (Figure 6.2) shows comparably little variation between individual plates. It is evident from Figure 6.3A...
(only channel 2 is shown) that for a number of plates one of the two replicates was removed, probably due to quality problems. Figure 6.3B shows that the correlation between the two replicates is satisfactory. Figure 6.3C shows that the positive controls mostly demonstrate a robust effect while the negative controls vary considerably. This effect is mirrored in Figure 6.3D, since the SSMD between the negative controls (exemplary shown for lrp5) versus the positive controls varies strongly from plate to plate. The SSMD for rela is between -3 and -6 qualifying as an acceptable control, while tnf1 is mostly between -3 and -6 and occasionally < -6. The sicon negative control appears to produce an elevated signal in channel 2 indicating that it may be affected by TNFα signaling. These findings are in agreement with Nickles et al., who disregard sicon as a negative control for the same reason. Moreover, the authors use the Z'-factor to test the positive controls with the findings that rela is unacceptable and tnf1 is acceptable as positive control. Figure 6.3E shows the column means across different plates with the typical u-shape indicating positional effects. A similar but less pronounced effect is observed for rows. In conclusion, the inconsistent performance of the controls suggests to use a plate based normalization method. Moreover, the pronounced positional effects suggest to apply B-score normalization. The density plots of the two replicates indicate which of the plate-based normalization methods are effective. The robust z-score and B-score are most effective in achieving congruence for the two replicate distributions. The raw data appears skewed towards higher signal intensities, which is partially mitigated by the robust z-score and by the B-score. It is noteworthy that the density plots produced from B-score normalized data do not appear normally distributed. Based on these results, we chose to continue with z-score, the robust z-score and B-score normalized data, while the original publication used the z-score.

For hit detection (Chapter 6.3.3), we chose median absolute deviation (MAD) with a typical threshold of 2.5, which led to a reasonable number of 614 hits (3.5% of targeted genes) in channel 1 and 427 hits (2.4% of targeted genes) in channel 2. Subsequently, the differential screening setting was used to remove all hits from channel 2 that were also reported in channel 1. The remaining 241 hits (1.4% of targeted genes) are thus exclusive to channel 2 and potential TNFα modulators. To our surprise, only some of the 19 hits were validated in secondary screening and none of the five genes that were fully validated by Nickels et al. produced a significant effect in our results irrespective of the normalization method applied (Supplemental Table A.7). Nickels et al. selected hits for follow up studies based on stringent filtering and literature search. However, the authors did not perform a systematic analysis of the hits that might also help to identify promising candidate genes. We thus performed gene set analysis (see Chapter 6.3.5) on the 241 hits we obtained. All conducted tests, i.e. GSOA and GSEA in both Gene Ontology and KEGG pathways, led to similar results implicating RNA polymerase II
related functionality and ribosomal activity with very high significance (Supplemental Tables A.8 to A.11).

Gene set analysis provided compelling evidence that the hits we obtained were related to NFκB signaling. However, this approach does not lead to the identification of novel modulators. For this task, de novo pathway enrichment (Chapter 6.3.5.3) is a more appropriate strategy, since it does not rely on pre-defined gene sets and already established molecular processes. To this end, RNAice constructed an indicator matrix consisting of a single column in which genes that are a hit are assigned 1 and 0 otherwise. Subsequently, we subjected this indicator matrix to KeyPathwayMiner (KPM). With only a single column in the indicator matrix, case exceptions (L parameter) become irrelevant. Consequently, we selected INES as enrichment strategy with \( K = 1 \) to allow one additional gene in the extracted sub-graph. We chose BioGrid [128] as source network for gene-gene interactions, since it is one of the established standards in the field. The top 20 solutions that were reported appeared to be variations and subsets of the best solution, which is shown in Figure 6.6. It contains two known members of the NFκB pathway, one of which is the exception node added to the solution. The remaining members were not yet implicated in NFκB-signaling and thus appeared as promising candidates for follow-up studies.

**Figure 6.6:** The 241 hits were subjected to analysis with the de novo network enrichment tool KeyPathwayMiner. The enrichment strategy INES with \( K = 1 \) produced as best solution a sub-graph containing RELA and CSNK2A1, which are both part of the KEGG pathway for NFκB signaling (indicated by red squares). The red node corresponds to the exception node, while green nodes were identified as hits.
6.4.2.2 Genome-Wide Functional Genomics Analysis for Genes Regulating Sensitivity to Vorinostat

Falkenberg et al. performed a genome-wide RNAi synthetic lethal drug screen to identify vorinostat sensitivity genes in human colon cancer cells. In addition, they included a genome-wide miRNA mimics screen under identical assay conditions. The combination of a RNAi and miRNA screens offered a unique opportunity to study if the observed effect in miRNA hits can be reproduced by the miRNA high-confidence target genes in the RNAi screen. Both screens were split into a plus and a minus drug arm, where the minus arm served to identify genes that were lethal by knockdown alone. We focused on the plus drug arm, which was interrogated for cell death using two different readouts. A cell viability assay was used to assess rapid cell death, while a complementary caspase activity readout served to assess apoptosis, i.e. slower cell death. We downloaded the raw data of the primary siRNA screen (PubChem AID 743454) and the miRNA mimics screen (PubChem AID 743456). Neither the siRNA nor the miRNA screen included replicates or controls such that only the (robust) z-score or B-score could be considered. However, since no positional effect was observed, the robust z-score was chosen as normalization method. For the viability readout (rapid cell death), we chose a hit selection cutoff of 2.5 MAD, which resulted in 32 miRNA hits (3.6% of miRNAs mimics). Putative miRNA target genes from RNAhybrid.hsa were obtained with a p-value threshold of 0.05. Here, the p-value reflects the likelihood of the miRNA-gene interaction considering the free binding energy determined by RNAhybrid. This lead to 48,980 possible miRNA-gene interactions. The subsequent filter for high-confidence targets using the hypergeometric test (see Chapter 6.3.4.1) with a p-value cutoff of 0.05 after multiple testing correction produced a single gene, CSDC2, which was also identified as a significant hit in the primary siRNA screen by Falkenberg et al. We repeated the same procedure with the miRNA hits identified via the alternative readout of caspase activity (slow cell death through apoptosis) using a threshold of 4 MAD, which yielded 34 miRNA hits (3.8% of miRNA mimics). The resulting list of 39,498 putative miRNA-gene interactions was filtered as described above, but did not yield any significant results. Next, we relaxed the multiple testing cutoff to 0.2, allowing for more false positives. This lead to 127 high-confidence targets found in the cell viability screen, out of which 6 were significant hits in the primary screen. Moreover, a single gene, TRIM27, was found in the caspase activity screen, which, however, was not a hit.
6.5 Discussion & Conclusion

HTS is an indispensable tool for drug target discovery and for functional genomics. A major concern in the field is assay robustness [173]. In primary screening thousands of samples are included in experiments that span over dozens of plates with only few replicates per sample, if any. Data normalization and hit detection are utilized to remove known sources of variation such as plate or batch effects. However, validation of hits in secondary screening is costly due to an increased number of replicates and through the inclusion of other cell lines to be tested. Therefore, only a limited number of hits from the primary screen can be selected and consequently, it is imperative to select the most promising ones. Sophisticated methods for quality control, normalization and hit discovery can help to better accommodate the characteristics of a particular screen. Here, the user has the choice between control and plate based normalization techniques and also needs to decide whether to correct for positional effects or not. The availability of replicates and suitable controls motivates the choice of a suitable hit detection method. Here, the SSMD and Bayesian hit detection method are available as robust alternatives to the traditional thresholds based on SD or MAD. The R packages RNAither and cellHTS2 provide excellent features for data visualization and processing of RNAi data. HTML reports and a web service for cellHTS2 allow non-expert users to perform data analysis independently. However, to be able to make an informed decision about the best possible data processing strategy, a non-expert user needs to be able to directly compare the effect of different data processing strategies interactively and with as little friction as possible. RNAice facilitates this in an interactive web application, where parameters can be quickly changed without the need to repeat the entire analysis. Moreover, RNAice provides a consensus approach to combine different normalization strategies. While RNAice already supports a large number of normalization and hit detection methods, there are additional methods that should be considered and integrated in the future. Examples are the aforementioned quantile normalization, as well as Li-Wong rank normalization, loess regression, or bivariate regression, which are found in cellHTS2 and RNAither. Another noteworthy method that should be included is the spatial correction method implemented in SbacHTS [174].

An important aspect for assessing the quality of hits from the primary screen is systematic functional analysis. While this is mostly covered by the HTSanalyzeR R package, only RNAice provides full integration of its functionality directly in the web interface. RNAi screens are currently the most popular choice for HTS experiments in academia, yet there are other popular types of screens that are not covered sufficiently by existing solutions. CRISPR/Cas9 screens yield comparable data and can utilize existing workflows for data analysis. Other types of screens, however, require dedicated features.
The analysis of small compound screens in *RNAice* benefits from the integration of the drug target database *STICH* to identify genes of interest. Unfortunately, at the time of writing, no small compound screen was available to serve as an application case. While a large number of small compound screens have been deposited in public databases such as PubChem, they lack raw signal data as well as plate id and well location for each sample. Other databases such as ChemBank provide this information but do not contain the PubChem identifiers necessary to identify the compounds in the available drug target databases. In theory, chemical structures represented by SMILES notation could be used to identify matching compounds via structural similarity search. This approach, however, is computationally demanding and not feasible in an interactive web application such as *RNAice*.

To accommodate *miRNA* screens, *RNAice* provides access to putative and experimentally validated *miRNA* target genes for identified hits. This allows subsequent analysis with HTSAnalyzeR for gene sets and with KPM for *de novo* network enrichment. It should be noted that two recent studies indicate that GSOA and GSEA do not lead to valid results when applied to lists of *miRNA* target genes. This is likely due to the fact that *miRNAs* target a large number of genes and thus have a high probability to always target a large number of important pathways. Therefore, we suggested a probability filtering step for high-confidence *miRNA* target genes that were targeted by more *miRNAs* than would be expected by chance. To evaluate the results of this method, we analyzed a coupled genome-wide RNAi and *miRNA* mimics screen. Our hypothesis that *miRNA* high confidence target genes would also be active samples in the RNAi screen could not be proved, since only a single significant high-confidence target gene was found. Nevertheless, the identified gene was a promising hit in the primary siRNA screen. Relaxing the p-value cutoff for multiple testing from 0.05 to 0.2 increased the number of high-confidence genes to 128, out of which only 6 were hits in the primary screen (4.7% hit rate versus 3.4-3.9% hit rate in the two readouts of the primary screen). A possible explanation for the low success rate could be the polygenic nature of miRNA regulation, in which an effect is likely to be caused by down-regulation of several genes, whereas the primary siRNA screen is limited to observing the effect of a single gene at a time. Nevertheless, additional work in this direction is needed to refine the method, e.g. by improving the filtering strategy to produce and validate a larger number of hits. It would also be worthwhile to use only experimentally validated *miRNA* targets instead of predicted ones to test this approach.

As another application case, we selected a genome-wide RNAi screen aimed at identifying modulators of TNFα and NF-κB signaling. Here, we found that the robust z-scores computed by the authors differ from our results. We thus compared the data.

---

8http://chembank.broadinstitute.org/ (access date: 06/01/2015)
processing strategy and identified a possible explanation in the use of shorth and quantile normalization prior to calculating the robust z-scores. Since the signal distribution was indeed skewed, such a normalization might lead to a more balanced detection between promoters and suppressors in a screen. This has to be considered carefully, however, since the normalization might also do harm in cases where more hits are expected in one of the two extremes due to the experimental design. One example is a caspase assay, which serves to detect apoptosis and thus is likely to include more promoters than suppressors in an experiment aimed at killing cells. In general, the differences found between our hit list and the hit list reported by Nickles et al. highlight the impact of comparing different normalization strategies.

A subsequent functional analysis of the screening hits, which was not performed by Nickles et al., implicated known functions of the NFκB signaling pathway (Supplemental Tables A.8 to A.11). De novo network enrichment with KPM identified two central elements of this pathway, namely RELA and CSNK2A1. The latter was selected as an exception node, connecting other genes that were identified as hits and which are not known to play a role in NFκB signaling. This highlights the potential of de novo enrichment analysis with KPM to suggest promising candidates for secondary screening and functional characterization.

In conclusion, RNAice enables biologists with little or no experience in R to analyze complex HTS data. The user is supported with various plots to allow for informed choices about the appropriate data processing strategy. Sources of variation can be spotted and accommodated before the hit list is produced. Moreover, different normalization methods can be compared conveniently. As a novelty, RNAice specifically supports the analysis of miRNA and small compounds screens, making RNAice the most comprehensive open-source tool for HTS data analysis currently available.
Chapter 7

MIRACLE: Microarray R-based Analysis of Complex Lysate Experiments

Objective of this Chapter

Development of a sample management and data analysis tool for reverse-phase protein array (RPPA) data, including

- virtual spotting to track samples from plates to RPPAs.
- user-friendly sample editor.
- quality control and visual inspection of the data.
- normalization of raw data to account for uneven staining.
- quantification to obtain single protein concentration estimates from dilution series.
- normalization for total protein amount.

While high-throughput screening (HTS) is an ideal strategy for the identification of potential drug targets, it provides limited insight into the underlying biological mechanisms. A secondary readout on the protein level using reverse-phase protein array (RPPA) is suitable to increase the amount of information gained from a single HTS experiment manifold. RPPAs are nitro-cellulose covered glass slides that can carry several thousand of samples of tissue or cell lysates. Since RPPAs are typically spotted in many copies, each copy can be stained with a different antibody, which is subsequently
Chapter 7. **MIRACLE**

linked to a biotin-complex. This enables fluorescence based measurement of protein expression levels \[60\]. See Chapter 1.7 for more details. Following image acquisition with a suitable scanner, commercial tools such as MicroVigene® can be used to measure the foreground and background signal intensity for each spot on an RPPA. The position of a spot is typically recorded as the combination of block number, row number, and column number, where each block originates from a single pin of the extraction head of the spotter. The extraction head has typically far fewer pins than there are wells on the source plates. Consequently, several extraction steps are necessary to transfer the samples of an entire plate onto a slide. Moreover, the distance between individual pins is relatively large, reflecting the distance of the wells on the plates. To fit as many samples as possible on a slide, the space between the pins is utilized by depositing subsequent extractions of the source plates with a minimal shift of the extraction head. In this way, samples that were taken up by the same pin will be placed next to each other, giving rise to the aforementioned blocks on the slide. However, with this strategy, the original order of samples in the source plates is lost, making it difficult to re-locate the samples on the slide. This is further complicated by the fact that samples are often spotted in various dilutions and with varying number of depositions, meaning that a single sample is represented by several spots distributed across a slide. The entire spotting process is shown in more detail in Supplemental Figure B.8. In the following, we will present RPPA specific requirements R1-R17 in sample tracking as well as in data processing and analysis and evaluate to what degree available tools are suited to fulfill these (Supplemental Table A.12). Subsequently, we will briefly describe the methods employed in RPPA analysis and present MIRACLE, a web application that is intended as a user-friendly end-to-end solution for RPPA data. This chapter is mostly based on \[176\], in which MIRACLE was first published.

### 7.1 Requirements

#### 7.1.1 Sample Management

To deal with the challenges introduced by the spotting process, an efficient and user-friendly sample management solution is necessary. Ideally, such a system is open source (R1) and accessible through a graphical user or web interface (R2). Sample tracking should be supported on the level of both, the microtiter plates used as source (R3) and the RPPAs generated from them (R4). To increase productivity, slides and plates should be organized in experiments or projects (R5). This extends to the fact that typically each slide is spotted in several slide copies (R6), where each one is stained with a different antibody. Moreover, the same slide may yield several readouts that
need to be tracked (R7). It would be advantageous to also include plate readouts to be able to utilize RPPA technology as a secondary readout to HTS experiments (R8). To optimally track samples between the source plates and the RPPAs, the sample spotting process should be modeled in silico in so-called virtual spotting (R9). To facilitate this, an appropriate RPPA sample tracking software must be customizable with respect to the plate format being used and the size of the extraction head, which may vary from spotter to spotter.

7.1.2 Analysis

Initially, visual inspection of the data can help to determine quality problems (R10). After background correction, the signal of the spots needs to be further processed. The first step is typically the correction for bias introduced through uneven staining (R11) [177]. Another concern is the dynamic range of signal detection, which can be described as a sigmoidal curve due to limitations in sensitivity in the lower range and signal saturation in the upper range [178]. Through adjusting each sample for the total protein amount a priori, measurement is possible in the linear range of this curve. However, this is often not feasible for high-throughput experiments, due to the trade-off between large sample numbers, feasibility, and costs. In order to overcome this problem, samples are typically spotted multiple times in a dilution series to cover a broad dynamic range of protein concentrations, where each sample gives rise to a response curve. In a process called quantification (R12), an estimate of relative protein abundance is created by merging these values (Supplemental Figure B.9). The resulting relative concentration estimates still need to be normalized for the total amount of protein (R13), before they can be evaluated statistically to assess the significance of relative differences in protein amount (R14). In addition, it is also of interest to correlate these values to plate based readouts obtained, for instance, from HTS experiments (R15). Finally, it would be desirable to include support for timecourse studies (R16) and network analysis (R17).

7.2 State of the Art

A single RPPA experiment may comprise thousands of samples distributed over large slide sets. Precise sample tracking is a challenge that grows with the number, size and complexity of the RPPA experiments. To date, the only documented solution to address this critical issue is an integrated platform called RIMS [179], which provides features for uploading and annotating sample information, data visualization, correlation and pathway analysis. Notably, the authors also propose an XML standard called RPPAML to overcome the lack of a data exchange format for RPPA data and a standardized
annotation. Unfortunately, however, none of the project URLs are accessible (last access attempt 06/01/2015), indicating that the project is no longer under active development and has not been adapted by the community. Being implemented for the commercial software MATLAB, RIMS also lacks integration of R methods commonly used for RPPA analyses. Finally, RIMS only supports sample tracking at the slide level and not at the level of the plate formats that form the basis for all experimentation. This leaves the most difficult step of sample tracking to the user.

Implementations for both, parametric and non-parametric methods for quantification of protein concentration estimates from serial sample dilutions are available through the R packages SuperCurve [180] and RPPAnalyzer [181]. These packages thus allow for the analysis of RPPA data. A major challenge here is, however, that end users are often not familiar with R. SuperCurve overcomes this problem partly by offering an tcl/tk-based graphical user interface, making both analysis and experimental design more accessible. Sample management on a larger scale, however, is neglected. RIMS addresses some of these issues, but does not cover more complex data analysis. See Supplemental Table A.12 for a feature comparison.

7.3 Methods

Before RPPA data can be used to compare protein expression levels of different samples, a number of processing steps are necessary, including data normalization, quantification and loading control normalization. The most common methods have been implemented in MIRACLE and are briefly introduced in the following.

7.3.1 Raw Signal Normalization

As customary for all microarray data, the background signal is determined for each spot and subtracted from the foreground signal. This approach, however, does not correct for signal bias due to uneven antibody staining, which is an issue specific to RPPA technology. Positive control spots on the slide can be utilized for creating a smoothing surface mirroring the staining bias. A correction factor can then be calculated from a generalized linear model for each individual spot (Supplemental Figure B.11) [177].

7.3.2 Quantification

Mircean et al. proposed a linear model [69] for merging the signal originating from individual response curves. One drawback of this approach is that a linear model cannot
deal with samples close to saturation or close to the detection limit. Consequently, Tabus et al. compared a variety of parametric models and found that a logistic model was most suitable to reflect the sigmoidal shape of the response curve. Furthermore, a joint response curve based on all samples increased the confidence of the model fit, since similar chemistry can be assumed for all samples [178]. Hu et al. showed that a more flexible non-parametric model yields more accurate estimates at the cost of robustness [180]. Finally, Zhang et al. proposed a simplified robust parametric model called serial dilution curve based on the Sips model for DNA binding. In contrast to other parametric models, this method is based on meaningful and intuitive parameters like the detection limit, the dilution factor and the saturation level [182].

7.3.3 Loading Control Normalization

If the total protein amount of each sample is not determined \textit{a priori}, protein levels have to be normalized, in order to guarantee a meaningful comparison between samples. This can be achieved by either normalizing to a slide stained for total protein with, for instance, Sypro Ruby [60] or Fast Green [183], or by using additional antibody stainings. For the latter approach, one can rely on either a selection of housekeeping proteins that are assumed to be constantly expressed or on incorporating the entire panel of antibody-generated signals, where all antibodies are first centered and scaled before the median value is used for normalization. This so-called median loading normalization has been improved by Neeley et al. in a method called variable slope normalization [184]. Here, a correction factor is included to take into account that additional bias arises due to independent slide measurements.

7.4 Results

The lack of a comprehensive and user-friendly solution for \textit{RPPA} sample management and data analysis motivated us to develop \textit{MIRACLE}, an open-source web application providing an end-to-end solution covering experimental design, sample tracking, data processing, normalization, as well as visualization and statistical analysis of the results. \textit{MIRACLE} conveniently keeps track of sample information, starting with the source plates, throughout array generation and down to the signal data, in a process called virtual spotting.

\textit{MIRACLE} allows biological researchers without any knowledge of \textit{R} to process and analyze \textit{RPPA} data efficiently, grasping back to established methods by interacting directly with \textit{R} in the background. This interface will also allows future methods to be added...
Figure 7.1: Schematic exemplary work-flow of RPPA construction and analysis via MIRACLE. (A) Larger sample sets are stored in multiple source plates, with individual and partly complex sample information. Optionally primary plate readout data can be included. (B) The individual sample lysates are diluted (indicated by the color gradient) in a first reformatting step and subsequently spotted onto slides in a customized pattern. (C) Multiple array copies are generated and stained with different antibodies, adding to the sample tracking demands. (D) An array scanner yields signal intensities for all spots, which need to be further processed to obtain the final results. Taken from [176].

in a straight-forward fashion. Results are directly visualized and can be investigated interactively with regards to statistical significance, as well as to correlation to primary plate based readout data.

MIRACLE is designed with user approachability in mind, but also supports R data analysts by offering a convenient data export/import interface with R. While the data analysis part of MIRACLE is particularly laid out for handling RPPA data, the sample management functionality is suitable for any kind of customized array design.

With its deep integration of sample management and data analysis, MIRACLE separates itself from existing solutions (Supplemental Table A.12) that only cover parts of the RPPA work-flow shown in Figure 7.1.

7.4.1 Sample Management

7.4.1.1 Plate and Slide Layouts

In a typical RPPA experiment, lysate samples are stored in a 96-well or 384-well microtiter plate, before they are subjected to microarray generation using a bioarrayer,
-printer or -spotter. Already at this stage, MIRACLE supports experimental design by offering an interactive web interface for creating so-called plate layouts. To avoid cryptic and long sample names, several layers of information can be included, for instance regarding cell material, treatments, applied compounds, lysis conditions, etc. (Figure 7.2). A lot of this information is shared by samples and is therefore redundant. MIRACLE stores information in relational databases, where each layer corresponds to a single property, keeping the data concise through use of ids and mapping tables. By relying on linked tables of a relational database, changes of layout properties are immediately available to all samples and experiments, thereby ensuring data consistency and comparability in the analyses. MIRACLE shares plate and sample related functionality with SAVANAH (see Chapter 8.1.3.3 for details), allowing both applications to remain compatible to each other and to share a single database. The latter guarantees that sample and plate information entered in HTS experiments can be conveniently reused in RPPA experiments and vice versa.

Similar to plate layouts, users can also define slide layouts, in which sample properties for each individual spot of the slide can be edited using the aforementioned sample layers. The format of the slide is determined through specifying the number of rows and columns, as well as blocks where applicable, e.g. when using spotters, where each pin of the extraction head gives rise to a different block.
7.4.1.2 Virtual Spotting

As previously mentioned, it is not always trivial to determine the location of a sample on the slide, since a large number of samples originate from different plates. Furthermore, repetitive spotting of samples with various dilutions and varying number of depositions per spot has to be considered, as well as the format of the extraction head. MIRACLE addresses this issue in a feature called virtual spotting, where previously created plate layouts are combined with information about the operation mode of the spotter, such as format of the extraction head, column or row-wise extraction, top-to-bottom or left-to-right spotting, to determine the final layout. The selection and order of the plates can be manipulated via drag-and-drop and for each plate individual extractions can be excluded. If a so-called deposition pattern is used, in which several adjacent spots originate from the same sample, but are spotted with varying depositions, the layout can be simplified. Since these samples are otherwise identical, the respective columns of the layout can be merged.

7.4.1.3 Projects and Experiments

MIRACLE enables users to locate sample information quickly using full text search. However, to keep experimental data organized, projects and experiments can be created, linked to layouts and subsequently be used for filtering. As mentioned before, the shared database between MIRACLE and SAVANAH (Chapter 5) allows projects and experiments to be defined in either of the two applications, thus avoiding overhead and ensuring that data stemming from related high-throughput experiments is properly linked to each other.

7.4.2 Data Processing

7.4.2.1 Slides

Slide layouts can be linked to an arbitrary number of slides, which correspond to the copies created during spotting. For each slide, additional information such as a barcode, the antibody that was used for staining and scanner settings, such as the wavelength of the readout, can be specified. Three types of files can be uploaded, including the output file from the scanner containing signal intensities, an image of the slide and an experimental protocol.
7.4.2.2 Supported File Formats

The experimental protocol can be of any file type (e.g. DOC, PDF or TXT), while for images the most common file types, such as JPG, PNG and TIFF, are supported. MIRACLE processes each image into the Microsoft® deep zoom format. In this format, an image is split into a number of tiles that provide the same resolution on a smaller part of the original image. These files are stored individually and split continuously until several levels of these splits are available. This leads to a hierarchy of tiles with the advantage that images can be zoomed to detail without requiring the entire image to be loaded at the full resolution. This is ideal for visual detection of quality issues, such as clogged tips, scratches or uneven stainings and also allows MIRACLE to efficiently handle the large image files of more than 100MB that are normally produced by array scanners. With regards to the numeric output of the array scanner, MIRACLE is not limited to certain file types, but has a flexible system supporting import of comma, semicolon, tab-separated or Microsoft Excel® files without requiring a specific format. This is facilitated by allowing the user to map columns of the input file to known properties in MIRACLE. This mapping can also be stored as a so-called ResultFileConfig.

7.4.2.3 Processing Raw Data

After successfully reading the scanner file, MIRACLE will offer to add all spots to the database. During this process, the signal of each spot is linked to the sample information stored in the slide layout. Subsequently, users can create heatmaps to visualize the data to detect quality problems. One example are block shifts introduced by the scanner software that can then be corrected for (Supplemental Figure B.10).

7.4.2.4 Plates and Readouts

Similar to how slides can be added to slide layouts, plates can be added to plate layouts, where additional information, such as plate and well type, barcode and replicate number are stored. Often a plate readout is performed spotting, using, for example, fluorescence-based measurement of cell viability or colorimetric analysis of total protein amount. MIRACLE allows for adding these results for each plate, utilizing the aforementioned file upload mechanism. This mechanism is identical to the one used in SAVANAH (Chapter 5) and allows for uploaded plate readouts to be used in either of the systems.
7.4.3 Data Analysis

When sample management and data processing are complete, users can begin with data analysis. After selecting a slide layout, the user is presented with a list of all slides linked to this layout. In case the slide layout was created through virtual spotting, readouts linked to the source plates are also shown. By starting the analysis, the user will be forwarded to an R-based web application called \textit{Rmiracle}, which will automatically begin to fetch the selected slide and plate readouts from the database.

7.4.3.1 Processing of Raw Signal:

\textit{Rmiracle} offers data analysis in several steps (Figure 7.3):

- A heatmap for visual inspection and correction of block shifts.
- Positive control spots can be used to correct for uneven staining using the method proposed by Neeley \textit{et al.} [177].
- If a dilution series has been spotted, a quantification method can be selected for merging these samples. \textit{Rmiracle} currently supports \textit{SuperCurve}, as well as implementations of a logistic model [178], serial dilution curve [182] and a non-parametric model [180].
- Slides can be normalized for total protein amount by selecting between median loading, variable slope [184] or housekeeping normalization. For the latter, one or several of the slides have to be marked as loading controls.
- Significance of relative sample differences can be assessed by selecting a sample reference for performing Dunnett’s test [185].

7.4.3.2 Protein Concentration Estimates and Sample Grouping

With the above settings, \textit{Rmiracle} computes protein concentration estimates that allow assessment of relative differences between samples. To this end, we grasp back to the multi-layer sample information model of \textit{MIRACLE} to group samples. Users can select horizontal and vertical grouping categories, for example cell lines tested or treatments applied, which will then be reflected by different facets of a bar plot. Users can also select an additional category called “fill” for separating bars by color to achieve a third grouping dimension to compare, for instance, replicates with different numbers of depositions. The results are also shown in tabular form, including a download option, and are
Figure 7.3: Processing of raw data in Rmiracle: Signal intensities are displayed in heatmaps (A) for visual inspection. Subsequently, a surface adjustment based on control spots may be performed to correct for uneven staining (B). In case of serial sample dilutions, quantification can be applied (C) to obtain a single protein concentration estimate (D). Furthermore, data can be normalized to negative controls (A_NC1) and total protein amount, e.g. using protein data from separate slides, which enables the identification of effector samples (E, depicted by arrows added to the plot). Significance of the sample differences is finally assessed in comparison to a selected negative control by applying Dunnett’s test (F). Taken from [176].
further accompanied by diagnostic plots specific for the selected quantification method. Figure 7.4 depicts the user interface of the \textit{Rmiracle} analysis. Beyond data visualization and computation of protein concentration estimates, the analysis comprises additional features introduced below.

### 7.4.3.3 Comparison across Slides and Readouts

The comparison tab provides a bar plot (Figure 7.4F), in which an average is calculated for the previously selected color fill category, since colors are here reserved for comparing protein concentration estimates across slides. It is also possible to include plate readouts.

### 7.4.3.4 Correlation with Primary Readout

An important aspect of quality control is signal correlation. Since we expect the raw signal and protein concentration estimates to follow a normal distribution, we use the Pearson correlation coefficient in the correlation tab (Figure 7.4G) to compare both, protein concentration estimates and raw signal intensities. These are presented as a heatmap and are also shown between slides and plate readouts. This can be an important factor, e.g. in case a plate-based readout provides information about the total protein amount and should therefore correlate with the \textit{RPPA} signal used for normalization.

### 7.4.3.5 Significance Testing

The significance of relative differences between samples or between samples and a control is of great interest for experimental researchers. Traditionally, t-tests are used to obtain the necessary p-values, often neglecting multiple comparisons correction and issues arising from low replicate numbers. To address these issues, \textit{MIRACLE} applies Dunnett’s test (Figure 7.4H), which is a t-statistic based multiple comparison method designed to keep the family-wise error below a threshold $\alpha$. In Dunnett’s test, each sample is compared with a pre-defined control. In contrast to other methods, the variance is pooled across all samples, thereby dealing with low replicate numbers [185].

### 7.4.3.6 Import and Export

Convenient import functions allow experienced \textit{R} users to download \textit{RPPA} data directly from \textit{MIRACLE} by specifying ids or barcodes, respectively. \textit{R} methods to process or visualize these data are directly available, allowing data analysts to perform deeper
Figure 7.4: Typical analysis using *Rmiracle*: A heatmap visualizes different slide properties, such as signal intensities (A). Users can change various parameters, for example inclusion of surface adjustment, selection of methods for quantification and normalization for total protein amounts. Specific samples can be selected and grouped based on different properties of the data set (B). Depending on the quantification method, diagnostic plots are shown (C). The results are displayed in an interactive table and in a bar plot (D,E). A global overview of protein concentration estimates is available for all slides and plate-based readouts (F). Pearson correlation coefficients are calculated for all slides, as well as for plate-based readouts (G). Significance is assessed by comparing sample groups to negative controls through Dunnett’s test (H). Taken from [176].

analysis not covered by the proposed work-flow. Each slide, as well as the resulting protein concentration estimates can be downloaded as tab- or comma-separated file, in which all layout information is included.

7.5 Discussion & Conclusion

*RPPA*s are a promising technology that finds growing application in both, basic and clinical research. While many of the challenges of this technology are similar to those of traditional microarrays, *RPPA*-specific challenges arise and have to be addressed. Especially when considering this technology as secondary readout to high throughput genome-wide *RNAi* screens, comprehensive tools incorporating all necessary tasks, such
as experimental design, sample tracking and data analysis, are essential. To fill this gap, we developed MIRACLE, a web-based application with deep integration of R for efficient data analysis.

With regards to sample management, both the SuperCurve package and RIMS provide a graphical user interface for specifying slide layouts, but it does not address sample tracking from plate to slide level and does not allow for multiple levels of sample information. Moreover, they lack the virtual spotting and layout editing features of MIRACLE that enable researchers to enter all sample-related information already on the plate level and before the complexity of the layout is increased by the array generation. This saves a significant amount of time and effectively avoids mistakes due to manual data processing.

The R packages SuperCurve and RPPanalyzer provide experienced R users with a wealth of options to analyse RPPA data. The results of these methods are relative protein concentration estimates. A logical next step could be to investigate how significant relative differences in protein levels are and how well results correlate, e.g. between slides used for normalization or between individual slides and plate-based readout. Only RPPanalyzer reports on slide to slide correlation [181]. Significance analysis is not part of any existing solution. Moreover, replicates are typically merged during data processing to increase confidence of the results, thereby making them unavailable for subsequent significance analysis. In contrast, Rmiracle processes replicates individually and offers a comprehensive evaluation of significance and correlation. A number of analysis methods, such as serial dilution curve [182] have been published as R code, but have not been adapted to a user-friendly format, thereby limiting their application for experimental researchers. Experienced R users, on the other hand, need the flexibility of the R environment to perform deeper analysis of the data. Rmiracle strives to serve both target groups by incorporating a broad number of published methods on RPPA data analysis in both, command line and web interface. Additionally, Rmiracle can be used completely independently of the MIRACLE web application, requiring only a local installation of R together with the required packages. To this end, data stored in MIRACLE can imported on the R command line either via user authentication or using a unique access token generated for each slide.

Notably, the web application RIMS followed similar goals as MIRACLE, but did not include scenarios where more sophisticated data processing, e.g. quantification and normalization of the signal intensities, is necessary [184]. Moreover, RIMS is not actively developed or available at the moment, stressing the need for a solution like MIRACLE.

In recent years, the application of RPPA technology has matured considerably. Along with this progress, suitable computational methods have been developed to address
issues in data processing. To further promote acceptance of this technology, fully integrated tools like MIRACLE are indispensable. Furthermore, it can be expected that standardization of RPPA data in a common framework can substantially aid the development of novel algorithms and allow better integration at the level of network biology and other multi-OMICS data. Finally, the integration of SAVANAH and MIRACLE provides a solid basis to couple two effective high-throughput methods, namely HTS and RPPA, not only on the experimental side, but also on the data management side. This reduces unnecessary effort for entering sample information and guarantees consistency. Finally, linking samples across OLF, SAVANAH and MIRACLE allows for these tools to be used efficiently in a drug target discovery platform.
Chapter 8

Implementation

Objectives of this Chapter

- Application development using the Grails web application framework.
- Concept behind the modular design of OpenLabFramework.
- Use of web services and secure tokens.
- Digital signature mechanism implemented in OpenLabNotes.
- Integration of SAVANAH and MIRACLE through a shared plug-in.
- Application development in R shiny.

8.1 Grails Applications for Sample Management

OLF (Chapter 3), SAVANAH (Chapter 5) and MIRACLE (Chapter 7) were built based on the same technologies and concepts, which are introduced in the following.

8.1.1 Grails Web Application Framework

We considered the Java™ based framework Grails\(^1\) to be the most promising candidate for building a web-application. Grails is an open source project and former VMWare® product that was built on the company’s experience in industry-standard frameworks such as Hibernate\(^2\) and Spring\(^3\), which also form the core of Grails. A plethora of

\(^1\)https://grails.org/ (access date: 06/01/2015)
\(^2\)http://hibernate.org/ (access date: 06/01/2015)
\(^3\)https://spring.io/ (access date: 06/01/2015)
open source plug-ins are available to deliver high-quality solutions for non-trivial web application tasks, e.g. database search (Compass and Apache Lucene\(^4\)), spreadsheet im- and export (Apache POI\(^5\)) and a user / security management (SpringSecurity\(^6\)). Grails embraces the paradigms convention over configuration and separation of concerns to keep the code concise and clean. Furthermore, Grails hides the complexity of data persistence with an object relational modeling technique, which encapsulates all database interactions and models them through Java domain classes. These characteristics allow rapid application development and facilitate the integration of new features built on top of reliable and robust technologies. Grails follows an extended Model View Controller (MVC) model, which depends on the following types of classes.

### 8.1.1.1 Domain Classes

Domain classes are used to define the data persistence layer of the application, i.e. they represent the model in the MVC pattern. Domain class instances are objects that represent rows in a relational database table, where each property corresponds either to a column of the row or to an association with another table / domain class. Java collections such as sets or lists allow setting up one-to-many and many-to-many associations, which are handled automatically on the database level through mapping tables that are used for joins. Associating database tables in this fashion raises the question of cascading behavior. Consider an example, where a class Library has a one-to-many relationship with another class Plate. If an instance of Library is deleted, one needs to decide whether associated instances of Plate should also be deleted. This cascading behavior can be defined in Grails by adding a “belongsTo” property to the domain classes. If, for instance, the class Plate is declared with a “belongsTo Library” statement, all associated Plate instances would be deleted along with the Library instance.

### 8.1.1.2 Controller and Service Classes

In the MVC pattern, the controller acts as the interface of the model and the view and implements the application logic. Grails deviates slightly from this pattern by splitting this functionality into two parts. Controller classes accept user input from the view and access the model to prepare and deliver a response. However, anything more complex than a simple information retrieval should ideally be outsourced to so-called service classes. This has the advantage of clearly separating the negotiation between model and view from the application logic, which in this way also becomes reusable and

\(^4\) http://www.compass-project.org/ (access date: 06/01/2015)
\(^5\) https://poi.apache.org/ (access date: 06/01/2015)
\(^6\) http://projects.spring.io/spring-security/ (access date: 06/01/2015)
can be shared by different controllers. Moreover, the default behavior of a service is transactional, i.e. any database manipulation will roll back if something goes wrong, thus protecting the integrity of the database.

8.1.1.3 Groovy Server Pages

*Grails* relies on Groovy Server Pages to realize the view part of MVC. A view is created as a mixture of normal HTML code, which can also include java script, and java code. When preparing a response, the embedded java or groovy code is evaluated prior to finalizing the HTML document. Templates can be used to make parts of a view reusable. Moreover, layouts can be defined to factor out common parts of the view. In addition, a large number of *Grails*-specific or customized tags allow accessing data provided by the server. Any dependencies, such as CSS or javascript files, are defined on the server side as so-called resources, which then only need to be declared to be included in a view.

8.1.2 Application Security

All *Grails* web applications presented here utilize the *SpringSecurity* plug-in to restrict access to the system. A role based access model has been established in which general functionality is available to members of “ROLE_USER”, while administrative tasks are only accessible to members of “ROLE_ADMIN”. To define access rules, two different systems are utilized. *OLF* defines access rules for controllers and actions on the database level through so-called request maps. In contrast, *SAVANAH* and *MIRACLE* use annotations of controllers and actions to establish access rules already on the level of the source code. Either approach is appropriate, but annotations are easier to manage.

8.1.2.1 Data Export to R via Web Service

Data export to *R* is realized through *Representational State Transfer (REST)* web services, in which *R* scripts and applications can request data from a *Grails* application using HTTP requests. These can be submitted, for instance, using the *RCurl R* package. The server will reply by delivering the requested content in the light-weight *JavaScript Object Notation (JSON)* format, which can be processed in *R* using, for instance, the package *RJSONIO*. The efficient conversion between database content and JSON objects is facilitated with the *Jackson* library. However, as mentioned before, access to *Grails* managed content is restricted to authenticated users. To authenticate via R, users can use *Rcurl* to facilitate a user login. Example code is shown in Listing D.1.

---

7https://github.com/FasterXML/jackson (access date: 06/01/2015)
However, to facilitate direct integration of *R shiny* applications (Chapter 8.2.1) with *Grails* applications, an alternative access system had to be devised. To this end, all exportable domain classes are equipped with a security token, which is a randomly generated universal unique identifier that cannot be guessed. The same concept is found throughout the internet, where content is often shared with other users through secure links that are based on such identifiers. If a *Grails* web service receives a request, it will first check if the user is authenticated. Alternatively, the service will check if a security token has been supplied. These are generated automatically for exportable objects in *MIRACLE* and *SAVANAH* such as slides and plate readouts and are offered to authenticated users for export (Figure 8.1).

![MIRACLE](image)

**Figure 8.1:** *Grails* managed content can be exported to *R* via a web service. Since access to these resources is restricted, two access models are available. Either, the user is authenticated via username and password or a resource-specific security token is utilized.

### 8.1.3 Implementation Details

#### 8.1.3.1 OpenLabFramework

As illustrated in Figure 3.1, *OLF* has a modular structure, in which a back-end plug-in provides the necessary base classes, as well as user, project and security management. Content plug-ins can then add arbitrary classes and view templates, and integrate with other plug-ins. All plug-ins are finally merged in the front-end application, which utilizes the scaffolding mechanism of *Grails* to dynamically create views.

The two most important base classes are called MasterDataObject and DataObject (Figure 8.2). MasterDataObjects can be extended by classes representing master data. This
represents content that is maintained by system administrators, e.g. wildtype cell lines or vector systems in the OpenLabGeneTracker module, or freezers and storage locations in the OpenLabStorage module. More dynamic content is expressed through DataObject classes, which can be created and modified by regular users, e.g. genes or genetically engineered cell lines in the OpenLabGeneTracker module, or StorageElements in the OpenLabStorage module, which represent location data through a fixed position in an associated storage box, which is in turn associated with a freezer. This hierarchy makes OLF highly generic, but also allows fine-grained interactions between more specialized classes. Finally, any OLF module can serve as a content provider for other modules or the front-end. Templates for building additional tabs, menus, or add-ins can be added.

Figure 8.2: The back-end of OLF provides two classes MasterDataObject (MDO) and DataObject (DO) that are extended by the different modules. MDOs can only be created by administrators, whereas DOs can be created by any user. Existing DOs and MDOs can be combined freely by module developers in order to build complex hierarchies.

REST web service:

To allow other applications to access information maintained in OLF, a REST web service has been implemented (Listing 8.1). Here, DataObjects in OLF can be searched and extracted given that a so-called app access token is provided. These tokens can be generated in the admin area of OLF with an optional expiration date. Access to the web service is limited to secure https connections to prevent hackers from intercepting the tokens or the content.
Listing 8.1: Two REST web services are available in OLF. The first one allows for searching DataObjects matching a query in OLF (line 1). Optionally, the type of DataObject can be specified, e.g. gene, or cellLine. The second method can be used to request detailed information on a particular DataObject via its id (line 2). Both methods require an app access token that has to be provided to gain access.

8.1.3.2 OpenLabNotes

The use of base classes introduced in OLF enables modules such as OpenLabFileAttachments to accommodate the additional domain class NoteItem that is introduced by OpenLabNotes without requiring any changes. Vice versa, OpenLabNotes allows samples to be linked to DataObjects that are generated by modules such as OpenLabGeneTracker. Moreover, the NoteItem class is immediately recognized by the Compass search engine, enabling full text search on all database fields.

![Diagram of OLF integration with OpenLabNotes](image)

Figure 8.3: Integration of OpenLabNotes in OLF. To maintain its flexiblity, OLF keeps track of arbitrary domain classes that extend a common class DataObject. This allows for complex interactions between modules. OpenLabNotes, for instance, generates DataObjects representing experimental notes and links them to other DataObjects, such as samples generated by OpenLabGeneTracker. The OpenLabFileAttachment module can generate DataObjects encapsulating uploaded documents and links them to either DataObjects irrespective of whether they represent samples or ELN notes.
Digital Signatures:

ELNs are expected to comply with regulations for effective data protection. This implies that access is generally limited to authorized users. In addition, documents need to be digitally signed to guarantee origin, authenticity and integrity. Hash functions, such as SHA-1, can be used to generate unique identifiers of an input string representing a document. Hash functions are one-way functions that produce a different output if the input string is changed. In this way, scientific fraud can in principle be detected. This, however, does not establish a sufficient level of protection, since the hash string in the database can simply be recomputed based on the modified document. Thus, the hash string is protected through asymmetric public-key cryptography (Figure 8.4).

First, a public/private key pair needs to be generated for each user. When a user signs a document, the private key, together with a password, is used to generate a digital signature. Subsequently, a public key can be used by any user to decrypt the digital signature and to validate the integrity of the signed document. Digital signatures are typically embedded in a document. However, both the author and a supervisor can sign a document, which would thus be stored redundantly in the database. We thus decided to generate detached signatures that are kept in addition to the original document. We apply the Digital Signature Algorithm (DSA), which is an officially recognized standard based on a variant of the ElGamal digital signature \[120\], which in turn is based on Diffie-Hellman key exchange method \[186\]. To add DSA capabilities to OpenLabNotes, the Bouncy Castle Java library\[^8\] was added. Apart from the document itself, it is imperative to protect timestamps, such as the creation date and the date of signing. Therefore, this information is included when creating the digital signature and for the validation.

8.1.3.3 Shared Functionality between SAVANAH and MIRACLE

Both, MIRACLE and SAVANAH deal with sample information and readout data related to microtiter plates. Therefore, shared functionality is factored into a separate Grails plug-in called HTSbackend (Figure 8.5). This does not only lead to better maintainability of the code, but also facilitates the integration of the two systems. This is desirable, since RPPA technology can be used as a secondary readout for HTS. Using a shared database, MIRACLE and SAVANAH can access and maintain the same information and users can enter and access plate level data in any of the systems. HTSbackend does not only provide domain classes, but also controllers and views, such that changes on any level are automatically available in both applications. Finally, HTSbackend enables the user to link cell lines created in MIRACLE or SAVANAH to cell line samples.

\[^8\]http://www.bouncycastle.org (access date: 06/01/2015)
Figure 8.4: Digital signatures require a pair of asymmetric keys that is generated by OpenLabNotes for each user. A: The author uses his private key, his password and a timestamp to sign a note. A second signature is added by the supervisor or a colleague based on his private key, the current timestamp and the timestamp of the author. The result is a detached signature for the author and the supervisor, respectively. B: The public keys of the author and the supervisor, together with the original note and the original timestamps can be used to validate the signatures.

Figure 8.5: The HTSbackend (middle) is a Grails plug-in that provides functionality shared between SAVANAH (left) and MIRACLE (right), such as project and experiments as well as management of plate level data.
8.1.3.4 Plate and Slide Layouts

As described in Chapters 5 and 7, sample data of plates and RPPAs is divided into layers and collected in plate- and slide layouts, respectively. To this end, plate layouts are associated with a collection of well layouts, whereas slide layouts are associated with a collection of spot layouts. Apart from its position, each well and spot layout is associated with a number of attributes corresponding to the various sample properties. Finally, all sample properties are independent domain classes and thus changes to a sample property are automatically propagated to all plates and slides that use it.

8.1.3.5 Plates and Slides

Similar to plate and slide layouts, plates and slides are associated with a collection of wells or spots, respectively. Each spot is linked to a spot layout and holds the foreground and background values of the measured signal intensity. In contrast, the measurement data of wells is stored in associated well readouts. Moreover, well readouts are represented as a collection in a readout, which in turn is associated with a plate.

8.1.3.6 Layout Editor

Experimenters typically use spreadsheets to track the sample layout of a plate. To provide them a similar or even better experience, the layout editor was developed. As shown in Figures 5.2 and 7.2, the user can select one of the sample properties that is then shown as a grid representing the plate or slide. Due to their size, slide layouts are split into groups of blocks represented by tabs. The grid is implemented as an HTML table combined with a java-script listener that monitors if a table cell is clicked on. The user can select several cells for updating a sample property by drawing a box. When the user presses the save button, the HTML table is submitted as a form, where table cells that were altered are used to update the corresponding database entries for well or spot layouts, respectively.

8.1.3.7 Grails OpenSeaDragon Plug-in

In order to efficiently deal with large image files in MIRACLE, we created the Grails OpenSeaDragon plug-in. The plug-in provides functionality to convert images to tiles according to the pyramide representation defined in the Microsoft® deep zoom format. Moreover, the OpenSeadragon java-script library is utilized to display these images providing a smooth zooming experience to the users of the system.
8.2 R shiny Applications for Data Analysis and Visualization

SAVANAH and MIRACLE provide sample management to HTS and RPPA data respectively. Their capabilities for data analysis, however, is limited to rudimentary data visualization. R is a language and data analysis software that is valued for its rich plug-in ecosystem, providing powerful packages for statistics and visualization. This makes R ideally suited to provide analysis and advanced visualization for HTS and RPPA data. A pitfall of R, however, is that its not very user-friendly. To overcome this hindrance, two web applications were developed in R shiny, namely RNAice for HTS data and Rmiracle for RPPA data.

8.2.1 R shiny

R shiny is an R package that enables developers to create web applications using R code. In this way, R packages and scripts used to analyze and visualize data can be transformed into a more accessible web interface. Moreover, R shiny applications can be deployed to a server, such that a user is not forced to maintain a local installation of R. This is particularly useful considering the complexity of package dependencies, which is a general issue for R users. The reactive programming paradigm underlying R shiny makes it particularly useful for setting up a data processing pipeline. In reactive programming, computationally expensive operations are encapsulated as reactive elements, which will only be recomputed if their input changes (Figure 8.6).

Figure 8.6: Reactive programming with R shiny. Changing Input B in the user interface will trigger a re-computation of data processing step 2, which depends on it. However, the potentially expensive computation in processing step 1 is only repeated if Input A is changed. Finally, the re-computation of processing step 1 will also trigger processing step 2, which depends on step 1.
8.2.2 Plotting

One of the most powerful plotting libraries available in $R$ is ggplot$^9$, which is used extensively in both RNAice and Rmiracle. Developing the analysis pipelines described above as web applications, however, suggests to utilize java-script to produce interactive rather than static plots to improve the user experience. To this end, several java-script libraries are available, such as highcharts$^{10}$ or polycharts$^{11}$, which were integrated through the rCharts$^{12}$ package.

8.2.3 Code Structure

$R$ shiny applications typically consist of merely two files. The ui.R defines the user interface and layout of the application, while the server.R is responsible for the application logic and for performing computations, producing plots, etc. The server.R consists mainly of reactive expressions that are coupled to each other, to input elements representing user interactions, and to output elements, such as plots or data tables. It can be imagined that developing a complex application with only two files results in code that is difficult to maintain. We realized this issue during the development of Rmiracle, which, however, is dwarfed in complexity by RNAice. This motivated us to devise rules for structuring $R$ shiny code that is logical and facilitates better maintainability in the development of RNAice.

First of all, the application logic in server.R is split into individual files. Each file delivers either a set of related functions (similar to methods in a service class) or a set of related output to be delivered to the ui.R. Functional code is located in a folder called server, while output generating code is located in a sub-folder server/output. Where applicable, complex code is factored out into separate functions located in a folder called source. Moreover, the ui.R is split into individual files that each describe the makeup of a single tab. These files are stored in a folder ui. Finally, all data files are located in a data folder, while additional java-script and CSS files are located in a folder named www.

Following this scheme, server.R and ui.R merely import other files and thus remain concise and readable. This allows developers to quickly assess in which files specific functions are located, contributing decisively to the maintainability of the project.

$^9$http://ggplot2.org/ (access date: 06/01/2015)
$^{10}$http://www.highcharts.com/ (access date: 06/01/2015)
$^{11}$http://www.polychartjs.com/ (access date: 06/01/2015)
$^{12}$http://rcharts.io/ (access date: 06/01/2015)
Chapter 9

Discussion and Conclusion

Breast cancer is a global burden that affects women in developed as well as in developing countries. Despite of improvements in diagnosis and therapy, it is the most common cause of death in women [187]. Determining the optimal treatment strategy for a breast cancer patient is complicated by the heterogeneous nature of the disease. Cancer in general is characterized by a failure in cellular regulation and initially triggered by point mutations in important genes. These mutations, known as driver mutations, may occur in different genes participating in the same biological function and thus lead to a similar outcome. This makes cancer a disease of pathways rather than genes [6]. Moreover, a sub-population of tumor cells, cancer stem cells (CSCs) are implicated in treatment resistance, metastatic spread and tumor relapse (Chapter 1.2.3). In deed, the existence of highly resilient CSCs can explain many of the properties observed in tumors. Some commonalities that are crucial for cancer development in general have been established in the so-called hallmarks of cancer [7]. These are caused by mutations in cancer cells that affect a series of oncogenes, which become hyperactive and are essential for maintaining the unnatural growth pattern of the tumor. This phenomenon, called oncogene addiction, can be exploited for cancer specific treatment in synthetic lethality (Chapter 1.4). Knowledge about synthetic lethal target genes can be used in precision medicine (Chapter 1.3), which aims at optimizing the treatment of patients by taking the molecular properties of the disease into account.

In breast cancer, several clinically relevant subtypes exist that mirror the different stages of differentiation in the normal development of mammary tissue (Chapter 1.2.2). The relation of these subtypes to the CSC model remains unclear, but it can be suspected that the breast cancer subtype is mostly determined by the differentiation stage of the founder cell. Subsequently, some tumor cells may acquire CSC characteristics. A first step in applying the principles of precision medicine to breast cancer patients, is the
successful classification of breast tumors into clinically relevant subtypes.

9.1 Classification of Breast Cancer Subtypes by Combining Gene Expression and DNA Methylation Data

Breast cancer subtype classification is currently based exclusively on histopathological profiling and on gene expression profiling tests. However, the widespread availability of other types of OMICS data in public resources such as TCGA [67] suggests an integrated approach to improve the prediction performance of statistical learning models and to discover novel biomarkers. In an initial project in the frame of this thesis, we thus first explored how the combination of two types of OMICS data, namely gene expression and DNA methylation data, could help to improve the classification of breast cancer subtypes (Chapter 2).

Briefly, a machine learning strategy using random forests was applied to build three statistical models. A gene expression model, a DNA methylation model, and a combined model based on the feature sets of gene expression and DNA methylation data. The performance of these models was compared to a control model based on the PAM50 genes that served as a gold standard to define the subtypes of the TCGA samples. Moreover, the robustness of these methods was assessed using .632 bootstrapping and the accuracy was determined using the average AUC (AAUC), which is an appropriate measure in multi-class problems. As a result, the DNA methylation model was clearly outperformed by both, the gene expression model and the combined model. To our surprise, however, the combined model did not utilize a single feature from the DNA methylation data. This suggests that the DNA methylation data is not suited for this task and led us to speculations about the fairness of the comparison, since the class labels were already based on gene expression data. This is also demonstrated in the original TCGA publication on breast cancer, where it is shown that DNA methylation data aggregates into five clusters that do not overlap with the subtypes investigated here [67].

The features we identified as relevant had only a small overlap with lists of known breast cancer genes. This indicates that most of the known breast cancer genes might not be promising candidates for subtype classification, since they are likely a common factor across subtypes. Moreover, the discordance of features lists, where only a few genes such as ESR1 clearly stand out, suggests that the issue that many genes have a moderate but significant contribution to the classification performance, might not be fully mitigated by the recursive feature elimination strategy we applied.
Nevertheless, we consider further investigation in the use of DNA methylation data for breast cancer subtype classification worthwhile. It remains unclear, for instance, if an improved pre-processing of the data might not lead to better results. The DNA microarrays applied in this study contain an average of 17 probes per gene, which are typically, but not exclusively, located in the promoter region of the gene. The strategy of using the median probe signal as a representative value for each gene significantly reduces the size of the feature data, but might not always be appropriate in cases where the signal varies significantly between probes.

Moreover, alternative methods for classification might still benefit from features with moderate predictive qualities. A novel strategy currently explored by PhD candidate Nicolas Alcaraz\textsuperscript{1}, for instance, attempts to utilize pathways rather than individual genes as features for random forest. Here, DNA methylation might contribute to a model based on both data sets, since features are evaluated in combination, i.e. as gene sets, rather than individually.

Most importantly, we consider the definition of subtype labels via the gene expression based PAM50 standard as a major problem in the TCGA data set. We could already demonstrate that some of the labels might be incorrect, which might be potentially harmful to any attempt of including alternative OMICS data, such as DNA methylation, miRNA expression of copy number variation data in the classification task. Alternative independent data sets, in which subtypes have been assigned using histopathological methods, should be included for further investigations in this direction and for model validation.

Despite its moderate performance, the DNA methylation model results contained a number of interesting genes such as Sorcin, which warrants further investigation. In general, OMICS data are well suited to generate sets of candidates that can then be functionally characterized through functional genomics. Using advanced cell manipulation techniques to systematically study large sets of potential biomarkers, however, leads to a large number of samples and demands the use of a LIMS for sample management.

\section*{9.2 OpenLabFramework: A Laboratory Information Management System with Focus On Functional Genomics}

In order to retain an overview over large sample libraries typically found in nowadays laboratories, an efficient system for management and tracking of samples is required. A

\textsuperscript{1}Computational Biology Group, Department of Mathematics and Computer Science, University of Southern Denmark
large number of LIMS have been developed in the past, none of which, however, could fulfill the requirements we defined as essential for a productive LIMS in the field of functional genomics. We were thus motivated to develop OpenLabFramework (OLF) to ideally support this important research field in the construction and maintenance of large libraries of vector construct and genetically modified cell lines (Chapter 3). OLF covers most of the requirements any sample management system should fulfill. Most importantly, this includes a physical sample tracking solution that guarantees that researchers do not waste time trying to retrieve samples that were once deposited in an unknown location. Moreover, we implemented a mechanism for printing and reading machine and human readable barcode labels to solve the issue of unidentifiable samples. To keep related results and documentation retrievable, OLF allows files to be uploaded and linked to samples. To bring physical and virtual sample management closer to each other, OLF supports mobile devices, allowing researchers to update the sample status right away after dealing with a sample. Some of the requirements we identified were concerned with deployment and support. Even a perfect LIMS would be of no use if the system cannot be installed in a particular laboratory. Limitations in the availability of a suitable server or to a certain database management system thus should be overcome. Therefore, OLF supports a large variety of deployment scenarios and comes with a thorough online documentation. Finally, OLF strives to overcome the limitation of most LIMS to their respective field of research. To this end, it is built in strictly modular fashion and with abstract classes that allow arbitrary modules to interact irrespective of the type of sample data. The implementation of a REST web service allows for other applications to integrate sample information maintained in OLF and is a significant step towards an integrated drug discovery platform (Chapter 8.1.3.1).

Using a database-driven web application, such as OLF, for sample management offers a number of advantages. Due to relational tables, all data are kept in a concise and consistent format, where changes and updates are automatically propagated. In contrast to file-based storage solutions, no experimental information is lost upon turnover of laboratory staff and no problems arise from cryptic and inconsistent sample terminology. Collaboratively creating experimental data is significantly more convenient in web-based applications, since concurrency issues, such as file locks, can be avoided. In addition, all information can be located quickly, using full text search, which additionally increases efficiency.

OLF currently provides rudimentary data export capabilities for all types of samples and for a number of different file types. These reporting capabilities, however, could be drastically improved by introducing support for customized reports, where information from several instances is pooled. A well known library that could be utilized to this
end is *JasperReports*\(^2\). Moreover, the use of the mobile view dedicated to smartphones and tablets could be improved significantly by implementing a dedicated app for mobile platforms, such as Android or iOS. A current limitation of the system, for instance, is that barcodes have to be scanned with a third party app on the device, while a dedicated mobile app could accommodate this directly.

Despite our best efforts, the deployment of *OLF* remains challenging for biologists. Moreover, they can not easily adapt the system to their particular research requirements without the assistance of a software developer. It would thus be advantageous to devise a mechanism that allows administrators of the system to define new sample types on the fly in the running system. The use of the label printing mechanism is currently limited to Dymo\(^\text{®}\) label printers, which creates a quite strict and unwanted dependency on the products of this company. Therefore, support for alternative label printers should be added in the future. Finally, two requirements we proposed remain unfulfilled by *OLF*. We suggested that integrating data analysis tools would be advantageous to enable users to deal with uploaded result files appropriately. An example for this is the *OpenLabTaqMan* plug-in that is currently under development. In functional genomics, *quantitative real-time polymerase chain reaction (qRT-PCR)* is regularly performed to measure the relative expression level of genes in samples. The intention of *OpenLabTaqMan* is to process raw data uploaded by the user to provide the possibility to retrieve, visualize, and compare several of these result files conveniently within *OLF*. Finally, the requirement of audit-logging to better monitor changes to the entries in the database and to establish a sample history remains an open challenge. A general laboratory task that was not covered in *OLF* until recently is the experimental documentation in an *ELN*.

### 9.3 OpenLabNotes: An Electronic Laboratory Notebook Extension for OpenLabFramework

Electronic laboratory notebooks (*ELNs*) provide several advantages over *paper laboratory notebooks (PLNs)*, such as full text search and data security. However, a major concern in the transition from paper-based to electronic documentation of experiments is the legal perspective. For a long time, only *PLN* were accepted as legal documents suitable to protect the interest of researchers and their institutions in legal disputes over patenting. Meanwhile, regulations for electronic data storage exist, which require that information entered by the user is protected to prevent scientific fraud. While many *ELNs* and general note-taking tools are available, none of the existing free solutions deals with

\(^2\)[http://community.jaspersoft.com/project/jasperreports-library](http://community.jaspersoft.com/project/jasperreports-library) (access date: 06/01/2015)
Chapter 9. Discussion and Conclusion

this issue. This motivated us to develop OpenLabNotes, which is a module contributing ELN functionality to OLF. It includes a rich text editor to provide researchers the text editing experience they are accustomed to from popular office software. Moreover, documents can easily be shared between users and collected in shareable notebooks. In addition, documents can be exported to DOCX or PDF formats. As a unique feature, OpenLabNotes provides a sophisticated digital signature mechanism based on a state-of-the-art Java security library. This enables the author of a note and an appointed supervisor to use their secret passwords to digitally sign a note, which cannot be altered anymore after this point. A timestamp is included in each signature to protect not only the document itself, but also the time of signing, which would be crucial in a law case. Finally, being an integrated part of OLF offers additional benefits, such as the possibility to upload and link arbitrary documents or to link the documentation to the samples that were used in a particular experiment.

Some features are missing in the current version that we would like to implement in the future. The introduction of user groups would make it more convenient to share documents with several users. In addition, it would be advantageous to add an audit logging system that could track different versions of a document. This would allow for specific versions to be signed, while the document would remain editable for future versions, thus limiting one of the current limitations of the system. Furthermore, the digital signature only extends to the documents and embedded images, but not to files that are uploaded and linked. Extending the signature mechanism to these files would be beneficial to allow scientists to use these files as proof for their invention. Finally, an open issue is the exact legal status of documents maintained in OpenLabNotes. While the implementation is open source and utilizes an established security library as well as an accepted standard for digital signatures, it lacks formal approval by the Federal Drug Administration, which defined the guidelines we adhere to. Further effort is thus needed to achieve official recognition.

9.4 SAVANAH: Sample Management and Visual Analysis of High-Throughput Screens

Recent advances in high-throughput functional genomics techniques and in the automation and miniaturization of high-throughput screening (HTS) have enabled researchers to study complex diseases, such as breast cancer, in a systematic fashion [29, 38]. Moreover, the HTS technology is ideally suited for drug target discovery [188]. Here, genes can be identified, for instance, that cause vulnerability of cells subsequent to a particular treatment, or discover scenarios of synthetic lethality, which can be exploited to
specifically target diseased cells. HTS is typically performed using molecular libraries, which include, for instance, small compounds that potentially act as a drug, siRNAs or shRNAs to facilitate a gene knock-down (RNAi), sgRNAs to facilitate a gene knock-out (CRISPR/Cas9), or miRNA inhibitors or mimics to perturb cellular regulation on a higher level. The production and maintenance of such libraries is elaborate and costly. Such that they are delivered in a set of microtiter plates in highly condensed form to allow for several screens to be performed. Prior to a screen, these library plates need to be diluted in several steps. This leads to a challenge in tracking various plates at different dilution stages.

To enable researchers to tackle this challenge effectively, we have developed Sample Management and Visual Analysis of HTS (SAVANAH) (Chapter 5). SAVANAH allows molecular libraries to be added to the system in a file upload. Subsequently, users can browse the content of the library, search for specific samples, and create a series of virtual diluted copies. Each library copy contains also copies of all library plates together with their corresponding barcodes. These are automatically created to identify the physical plates unambiguously. A number of diluted libraries can be selected in a step-by-step guide in preparation of a screening experiment. These diluted libraries serve as replicates and are associated through a common plate layout, which is created from the respective library plate and additional experimental information provided by the user. Each plate can be linked to several readouts, which can be uploaded individually or as a batch. Finally, SAVANAH allows these data to be exported along with the sample information for further analysis in RNAice (Chapter 6). The convenient plate editor that displays sample information across several layers can not only be used to edit the plate layouts used in large-scale library screens, but is also ideally suited to manually design experiments in a smaller scale. This makes SAVANAH an attractive tool for HTS facilities as well as for laboratories that perform a large number of plate-based assays.

There are, however, several features that would make SAVANAH more applicable for certain use cases. While data processing and analysis is addressed by RNAice, it would be advantageous if researchers could upload the results of the analysis to SAVANAH, similar to how this is handled in competing tools such as Screensaver [125]. Documenting the analysis would allow for results to be retrieved more easily and would thus improve the overall documentation. Small compound screens would benefit from features utilizing the structural information known about the tested molecules, e.g. by including a visualization of the chemical structure based on the SMILES representation [175]. Moreover, the current focus on SAVANAH is either on primary screening or on small customized experiments. It would be advantageous if the user was assisted in secondary confirmation screens, where plate layouts need to be created based on a given hit list.
from the primary screen. Automated plate randomization could be offered to better accommodate clustered libraries. In addition, dedicated support of deconvolution screens would be beneficial, where samples that were pooled in the primary screen are tested individually. Information across different screens could be utilized, e.g. by monitoring the performance of individual samples to determine off-target effects or other quality problems \[189\]. To further increase the accessibility of the system, im- and export of plate layouts in XLSX or CSV format should be included. SAVANAH currently accommodates various types of high-throughput screening (HTS) data, but could in the future be extended to support high-content screening (HCS) data. Finally, SAVANAH could achieve broader applicability by explicitly supporting time course or dose response studies.

9.5 RNAice: High-Throughput RNAi and Compound Screen Evaluation for R

A single HTS screen may include several thousand of experiments, which are typically spread across dozens of microtiter plates and which are often seeded from different cell batches (Chapter 1.6). While the simple numerical output of most readouts can in theory be evaluated relatively easily, there are numerous technical and biological factors that introduce a significant amount of variation, which are reviewed in \[56\]. A straightforward solution to this issue is to increase the number of biological and / or technical replicates to reduce the variation of the experimental data to a tolerable level. Moreover, replicate plates can be randomized to reduce the effect of positional bias. However, the significant costs and efforts associated with these experiments do often not permit such strategies. In consequence, primary screens are often conducted in triplicates, duplicates or without replicates. Alternatively, one can rely on computational methods to normalize the data for known sources of variation to increase the quality of the results.

Several strategies have been proposed in the past, reaching from control-based normalization methods that utilize positive and negative controls placed on each plate. Since controls may not always be reliable and are also limited in replicate number, plate-based normalization is an attractive alternative, which operates under the assumption that most of the samples in a plate do not exhibit any effect and can thus contribute to a virtual negative control. Plate-based normalization methods can be simple, like the z-score, which takes only the variation of all samples into account, or more complex such as the B-score, which uses a two-way median polish to deal with row- and column-specific effects. To utilize replicate data, the SSMD is useful, since it considers the variation of the sample and the negative controls independently. The recognition
of hits can be based on any of these methods in combination with a fixed threshold, a multiple of the SD or MAD, or based on a Bayesian model. The latter utilizes negative control wells of the individual plates and across the experiment, as well as the variation of the sample itself.

The success of each of these methods, which can realistically only be assessed through the confirmation rate observed in secondary screening, depends on several factors. This includes the quality of negative and / or positive controls, the availability and agreement of replicates, and the magnitude of batch, plate, and positional effects. The choice of the appropriate data processing strategy is thus not trivial. This motivated us to develop RNAi and Compound Screen Evaluation (RNAice) (Chapter 6), a web application implemented in R shiny, which in contrast to existing analysis packages for HTS data provides a high degree of flexibility in comparing different data processing strategies. A rich visualization of raw data, hits, and possible indicators of variation, make RNAice the most user-friendly HTS non-commercial analysis software currently available.

In addition to its guidance in the hit discovery process, RNAice also supports the user in the down-stream evaluation of potential hits. This support extends to RNAi or CRISPR/Cas9 screens, small compound screens, and, as a novelty, to miRNA inhibitor and mimics screens. In practice, hits from small molecule or miRNA screens are mapped to putative or known targets genes. Consequently, each of the three screen types provides a list of genes that can be analyzed using gene set overrepresentation analysis (GSOA), gene set enrichment analysis (GSEA) or de novo network enrichment analysis in order to investigate these hits in the light of established biological knowledge.

Finally, RNAice allows to compare several readouts of the same screen or several screens of the same library with different experimental parameters, e.g. comparing different cell lines or to compare the effect of a specific treatment. Here, one of the screens or readouts needs to be selected as a reference, i.e. hits from this screen or readout are generally excluded from the hit lists. Subtracting such a baseline screen allows to remove hits with unspecific effects, such as general toxicity from hits with specific effects, such as a treatment response. This option is currently very rudimentary and provides several angles for future improvements. It should, for instance, be possible to select different normalization and hit detection methods for each of the considered screens, since each of them might be affected from different sources of variation to a varying degree. Moreover, the differences in signal distribution between different readouts and screens suggest turning towards rank-based solutions. This could be, for instance, the use of rank products [136]. Moreover, RNAice does currently not provide any means to effectively compare primary and secondary screening data. It should be mentioned that in absence of raw data from small compound screens, this particular use case could not
be tested appropriately. We are, however, currently in the process of requesting suitable
data from authors of corresponding studies.

The HTS community continues to propose new and effective normalization strategies
that can deal with known sources of variation. Such methods need to be included into
RNAice to allow for a comprehensive comparison. In this way, RNAice could in the
future not only serve as a data analysis tool, but also as a benchmark tool for method
development. In this way, it would allow for assessing the effect of various normalization
methods, such as quantile normalization. Finally, a major problem in the successful
adaption of RNAice is the absence of plate-level information from public databases such
as PubChem. This poses a significant issue in establishing the trustworthiness of pub-
lished results, which are often based on simple and potentially ineffective normalization
methods. In comparison, other research methods, such as microarray expression data,
have overcome this issue. Most journals now require that such data is deposited in
public databases such as GEO [190]. The availability of increasing amounts of HTS
data suggests to interrogate these data in conjunction through so-called meta-analysis.
However, the absence of raw data effectively prevents such studies.

9.6 MIRACLE: Microarray R-based Analysis of Complex
Lysate Experiments

In recent years, the application of RPPA technology has matured considerably. Along
with this progress, suitable computational methods have been developed to address
issues in data processing. This includes spatial correction of staining effects, the quan-
tification of signal data stemming from dilution series, and the normalization of protein
concentration estimates for the total amount of protein in a sample. However, the issue
of sample management has been widely ignored, preventing the successful application
of this technology for high-throughput scenarios.

This motivated us to develop *Microarray R-based Analysis of Complex Lysate Experiments*
(MIRACLE) (Chapter 7), which is the first open-source sample management system
dealing with bioinformatics challenges inherent to RPPA experiments. A common issue
of managing RPPAs data is the large number of sample spots for which several prop-
erties need to be tracked across dozens of slide copies. Moreover, individual slides can
be scanned with different settings, further increasing the number of data points. The
information for each sample spot typically includes details about the sample itself, its
treatment, and the estimated amount of cells, as well as spot specific information such
as its position on the slide, foreground and background signal, dilution, and number of
depositions. *MIRACLE* provides a slide layout editor that allows users to assign and modify these sample properties conveniently for an entire slide. To make use of the fact that samples with varying deposition numbers are typically spotted next to each other, *MIRACLE* asks the user to define a deposition pattern. In this way, several columns or rows of the layout can be merged, since the remaining sample properties are identical. This simplifies editing the slide layout, which is shared by all spotted copies. To further reduce the effort for defining a slide layout, *MIRACLE* supports management of plate level data. This functionality, which is shared with *SAVANAH* (Chapter 5), enables users to enter sample information prior to the spotting. At this point, the location of samples is easier to oversee for the experimenter, since the spotting process often leads to a seemingly randomized distribution of samples across the slide. This process is mirrored *in silico* by *MIRACLE* and referred to as virtual spotting.

Most available analysis packages for *RPPA* data are difficult to use for biologists without experience in *R*. To mitigate this, the *SuperCurve* *R* package provides a graphical user interface. The installation of *SuperCurve*, however, is extremely complex due to the use of non-standard repositories. The *R* package that is part of *MIRACLE* has similar issues, but can be deployed centrally thanks to *R shiny*. Here, data analysis is performed remotely on a server and accessed through a web browser. Moreover, *MIRACLE* facilitates the direct export of selected *RPPA* slide and accompanying plate readout data directly to the *R shiny* application, where it can be analyzed. This part of *MIRACLE*, called *Rmiracle*, also provides rich visualizations and a statistical evaluation of differences in protein amount between different samples. In conclusion, *MIRACLE* provides a seamless transition from sample management to data analysis and supports the user in every aspect of *RPPA* data handling. Possible improvements to *MIRACLE* include implementing options for im- and export of plate and slide level information directly from XLSX spreadsheets and the possibility to add the results of the analysis conducted in *Rmiracle* to *MIRACLE* itself to allow storing this information in the database for future reference.

To further promote acceptance of *RPPA* technology, fully integrated tools like *MIRACLE* are indispensable. It can be expected that standardization of *RPPA* data in a common framework can substantially aid the development of novel algorithms and allow for better integration at the level of network biology and other multi-OMICS data. The *Minimum Information About a Microarray Experiment (MIAME)* standard [191] and platforms like GEO [190] offer an effective method for standardized data exchange of gene expression array data. In GEO, users can export data to *R* or utilize a web-based application called *GEO2R*\(^3\) for basic analysis. *MIRACLE* has the potential to deliver similar features to the *RPPA* community in the form of an *RPPA* web portal for

\(^3\)https://www.ncbi.nlm.nih.gov/geo/geo2r/ (access date: 06/01/2015)
published data, similar to what has already been achieved for RNAi HTS through the GenomeRNAi database [192]. In addition, we envision that MIRACLE could serve as a framework for comparing the performance of various methods for RPPA data processing. Finally, it is imperative to add capabilities for easier down-stream analysis on the pathway level, for instance, by integrating with third party tools like RPPApipe [193].

9.7 An Integrated Functional Genomics Platform for Drug Discovery

Advances in cell manipulation techniques in functional genomics allow for targeted manipulation of the genome. With this, fine grained studies of wildtype genes and / or their mutated variants is feasible on a large scale. In high-throughput genomics, an entire panel of isogenic cell lines is created, which can be efficiently characterized using HTS. In this way, the effects of gene perturbation can be studied on the genome level, making such an experimental setup ideal for discovering disease vulnerability genes or to uncover scenarios of synthetic lethality. However, the information gained by HTS experiments is quite limited. While differential HTS can identify effects that are specific to the perturbed gene, it remains often unclear how this effect can be explained in the frame of established knowledge of a disease. In particular in cancer research, there are many known disease genes that could possibly be involved and deliver an explanation for the observed effect. Therefore, RPPA technology is an appropriate secondary readout to HTS screens that allows measuring the effect of genome perturbation on the protein expression of known disease markers. The combination of functional genomics, HTS and RPPAs thus constitute an effective platform for drug discovery.

A major goal of the project underlying this thesis was to develop the necessary bioinformatics tools to optimally support the three integral parts of the drug target discovery platform described above. OLF was developed to handle the sample management challenges of creating and maintaining large panels of isogenic cell lines and associated vector constructs. SAVANAH was created to deal with the challenges of dealing with molecular libraries used in HTS and to deal with the large number of diluted library plates and assay plates that are created in the process. To provide user-friendly access to the processing and analysis of such data, RNAice was developed. Finally, the development of MIRACLE allowed researchers to deal with the particular challenges of RPPA data that were introduced by the spotting process and by the necessity to spot samples in serial dilutions. Moreover, MIRACLE provides capabilities for analyzing these data in a user-friendly fashion. Another goal of this project was to integrate these tools to further increase productivity in sample management and data handling and in this way,
to complement the experimental drug target discovery platform with a corresponding integrated bioinformatics platform. This goal could be achieved through a number of mechanisms. The REST web service that was implemented in OLF is currently utilized by both, SAVANAH and MIRACLE to link plate based samples to cell line samples in OLF. Moreover, SAVANAH and MIRACLE share a common database and all plate specific functionality, allowing researchers to generate and access assay plate and readout data in either of the two applications. This also guarantees that plates produced in SAVANAH in an HTS experiment can be further processed in MIRACLE to produce slide layouts needed for managing RPPA data. In conclusion, this establishes a thread reaching from functional genomics, via the primary readout in HTS to the secondary readout in RPPAs.
Chapter 10

Outlook

A central goal of the project underlying this thesis was to develop user-friendly and effective tools for sample management and data analysis for functional genomics, HTS and RPPAs. These tools have the potential to create an impact in each of their respective research fields. A common issue found in all of the tools developed here, however, is the complex deployment, which might prevent other researchers from using these tools successfully. Grails applications can be delivered as self-contained and executable JAR files. The R shiny applications introduced here, however, require an R installation and depend on a large number of R packages originating from several repositories, including CRAN\(^1\), Bioconductor [194] and github\(^2\). In addition, it is quite likely that future updates of R and / or related packages may break essential functionality of these applications. In consequence, dependent tools such as SAVANAH and MIRACLE will also be affected. To mitigate this, we plan to release all of the software tools described here as self-contained Docker\(^3\) images.

Docker provides an abstraction layer of the operating system and thus a novel approach to virtualization. This allows running software through so-called docker containers, which are light-weight and particularly suited to deliver complex systems such as the ones described in this thesis. Containers are executed based on images, which include the software and all of its dependencies. In this way, the software can easily be maintained and updated through subsequent image releases. Moreover, Docker containers can share file directories and use network ports of the host system, allowing web applications to be deployed as if they were hosted in an ordinary way. Containers can also be linked, thus facilitating a modular deployment of the different parts of the drug target discovery platform introduced in this thesis. Finally, the light-weight abstraction model

---

\(^1\)http://cran.r-project.org/ (access date: 06/01/2015)
\(^2\)https://github.com/ (access date: 06/01/2015)
\(^3\)https://www.docker.com/ (access date: 06/01/2015)
of Docker provides nearly the same performance as a native installation, in contrast to the considerable overhead that would be introduced by a virtual machine approach.

Another open challenge is the integration of the data analysis tools, which should be further improved. A major goal of the drug target discovery platform described in Chapter 1.8 is to evaluate samples with significant effect in HTS in the context of the observed protein expression of disease markers. However, while Rmiracle, the analysis part of MIRACLE, can already deal with plate readouts, it does currently not benefit from the data normalization and hit discovery capabilities provided in RNAice. Consequently, further efforts are necessary to better link the results of these two tools. In addition, it remains unclear how these data can be optimally analyzed together. More research in this direction is necessary, yet a straight-forward approach would be to subject these data to de novo network enrichment to uncover the potential relationship between disease markers and the hit genes. More sophisticated techniques may be necessary though, such as the Bayesian network inference approach for RPPA data suggested by Hill et al. [195].

It can be expected that the drug discovery platform described in this thesis will be extended with additional high-throughput technologies in the foreseeable future. Some of these technologies already impact current research. It is, for instance, widely acknowledged that the development of targeted therapeutics requires a better understanding of within-tumor heterogeneity [196]. This is feasible through advances in NGS technology that enable single cell sequencing. This has great potential to unravel the mutation patterns leading to cancer cell differentiation and to CSC development [197]. Moreover, this technology can be used to study the properties of metastatic cells in the blood stream [198]. The sequencing data generated in this process will introduce new challenges in data management, processing and integration that have to be addressed. Moreover, recent advances in microfluidics allow HTS experiments to be performed on the single cell level [199]. The use of microfluidics technology will likely require the integration of new methods for data processing and sample tracking.

Established HTS applications such as RNAi screening might in the near future become possible through more sophisticated technologies that enable these experiments to be performed at a fraction of the costs. An example for this is pooled shRNA screening, in which NGS is used to perform genome-scale RNAi in a single experiment [75]. Even though the advance of such technologies challenge the traditional HTS platforms, it can be expected that the unique flexibility and throughput of traditional HTS will lead to new applications. This has recently been demonstrated by the availability of genome-wide miRNA inhibitor and mimics libraries. In conclusion, it is imperative that the bioinformatics drug discovery platform presented in this thesis continuously evolves to
adapt to new technologies and to new types of high-throughput applications.
Bibliography


Chapter 10. Outlook


Chapter 10. Outlook


[198] D. Ramskld et al. *Full-length mRNA-Seq from single-cell levels of RNA and individual circulating tumor cells*. 2012. DOI: 10.1038/nbt.2282.


List of Figures

1.1 Breast cancer subtypes can be linked to different stages of cellular development in mammary tissue. ................................................. 4
1.2 The cancer stem cell model. .................................................. 5
1.3 The concept of synthetic lethality. ......................................... 7
1.4 The RNAi pathway can be utilized for gene knock-down with siRNAs and regulates gene regulation through miRNAs .................. 9
1.5 Setup of the drug target discovery pipeline proposed in this thesis, which also implies the thesis structure ................................. 18

2.1 Overview of the different random forest models and the statistical learning strategy. ................................................................. 24
2.2 The .632 bootstrap error over 10 iterations. ............................ 26
2.3 The classification and out of bag error. .................................... 27
2.4 The 15 most important features determined by the mean decrease in gini index. ................................................................. 29
2.5 Venn diagram of the overlap between feature lists and the PAM50 genes. . 30
2.6 Venn diagram of the overlap of the feature lists and known breast cancer genes. .............................................................. 31

3.1 OpenLabFramework is built using a modular structure ............... 39
3.2 OpenLabFramework features an intuitive web-interface consisting of four parts ........................................................................... 42
3.3 A typical work-flow for OpenLabFramework ............................ 43

4.1 The show note view in OpenLabNotes. ........................................ 49

5.1 Screening library dilution scheme .............................................. 52
5.2 Plate layouts in SAVANAH .................................................. 55
5.3 SAVANAH visualizes readout data as heatmap or scatter plot. .... 56
5.4 Browse the wells in a screening libraries in SAVANAH ............ 57
5.5 Browse library dilutions in SAVANAH .................................... 58
5.6 User guide for creating an experiment using a library in SAVANAH . 59
5.7 Scheme depicting how libraries lead to assay plates. ................. 60

6.1 RNAice input tab ................................................................. 76
6.2 RNAice data tab .................................................................. 78
6.3 RNAice quality control tab .................................................... 79
6.4 RNAice hit discovery tab ....................................................... 81
6.5 RNAice consensus hits tab .................................................... 81
6.6 TNFα modulators identified by de novo network enrichment  
7.1 Schematic work-flow of RPPA construction and analysis via MIRACLE  
7.2 Sample management of plate layouts in MIRACLE  
7.3 Processing of raw data in Rmiracle  
7.4 Typical analysis using Rmiracle  
8.1 Export of Grails managed content to R  
8.2 The domain object structure in OpenLabFramework  
8.3 Integration of OpenLabNotes in OLF  
8.4 Digital signature strategy employed in OpenLabNotes  
8.5 The HTSbackend Grails plug-in provides shared functionality to SAVANAH and MIRACLE  
8.6 Reactive programming with R shiny  
B.1 All 53 selected features of the gene expression model  
B.2 All 38 selected features of the methylation model  
B.3 The top 50 features of the combined model that have a mean decrease in gini index > 2.  
B.4 The .632 bootstrap error over 10 iterations when excluding samples labeled as normal  
B.5 The classification and out of bag error of each subtype when excluding samples labeled as normal  
B.6 Example of a heatmap covering the entire screen in RNAice.  
B.7 Comparison of different plate based normalization methods in a screen for TNFα modulators  
B.8 RPPA spotting yields blocks of samples that correspond to the pins of the extraction head.  
B.9 To cover a broad dynamic range, samples are typically spotted in a dilution series on an RPPA.  
B.10 Correction of block shifts in MIRACLE  
B.11 Surface normalization in Rmiracle using positive control spots
List of Tables

2.1 Confusion matrices for the different random forest models. 28

A.1 Overview of existing LIMS solutions. 163
A.2 Feature comparison of LIMS with sample management support. 164
A.3 Feature comparison between the IARC ELN and OpenLabNotes. 165
A.4 Feature comparison between SAVANAH and comparable open source solutions for managing HTS data. 166
A.5 Screening libraries can be added to SAVANAH via tab delimited files with a specific header. 166
A.6 Feature comparison between RNAice and other open source solutions for analyzing HTS data. 167
A.7 The 19 suspected TNFα modulators that were subjected to secondary screening. 168
A.8 GO molecular function terms found to be significantly overrepresented or enriched in hits found in RNAi screening for TNFα modulators. 168
A.9 GO cellular compartment terms found to be significantly overrepresented or enriched in hits found in RNAi screening for TNFα modulators. 169
A.10 GO biological process terms found to be significantly overrepresented or enriched in hits found in RNAi screening for TNFα modulators. 169
A.11 KEGG pathways found to be significantly overrepresented or enriched in hits found in RNAi screening for TNFα modulators. 170
A.12 Features of MIRACLE and other open source solutions for processing RPPA data. 170

157
Abbreviations

AAUC average AUC. 25, 31, 32, 120, 175

AID PubChem Assay ID. 62, 75, 86

AUC area under the ROC curve. 25

CLIP crosslinking immunoprecipitation. 71

CRISPR clustered regularly interspaced short palindromic repeats. 10, 11, 65, 70, 73, 82, 87, 125, 127

crRNA CRISPR RNA. 10

CSC cancer stem cell. 1, 3–5, 14, 17, 119, 134

DBMS database management system. 37, 40

dsRNA long double-stranded RNA. 9

ELN electronic laboratory notebook. 19, 43, 45–50, 112, 123, 124, 157, 165

EMT epithelial to mesenchymal transition. 2, 4

ES enrichment score. 73, 74

esiRNA endoribonuclease-prepared siRNA. 9

FDR false discovery rate. 69, 70, 188

GLONE Global Node Exceptions. 75

GOF gain-of-function. 6, 11
**Abbreviations**

**GPLv3**  GNU Public License v3. 181

**GSEA**  gene set enrichment analysis. 73, 74, 82, 84, 88, 127, 168–170

**GSOA**  gene set overrepresentation analysis. 73, 74, 82, 84, 88, 127, 168–170

**HCS**  high-content screening. 13, 54, 126

**HTS**  high-throughput screening. 1, 2, 9, 12–19, 50–54, 59–62, 64, 65, 67–69, 73, 75, 76, 83, 87, 89, 91, 93, 97, 105, 113, 114, 116, 124–128, 130, 131, 133, 134, 157, 166, 167

**INES**  Individual Node Exceptions. 74, 75, 85

**JSON**  JavaScript Object Notation. 109

**KPM**  KeyPathwayMiner. 74, 75, 83, 85, 88, 89

**LIMS**  laboratory information management system. 12, 19, 35–39, 42–44, 47, 50, 54, 121, 122, 157, 163, 164

**IncRNA**  long non-coding RNA. 6, 12

**LOF**  loss-of-function. 6–11, 13, 14

**MAD**  median absolute deviation. 67–69, 84, 86, 87, 127

**MET**  mesenchymal to epithelial transition. 2, 4

**MIAME**  Minimum Information About a Microarray Experiment. 129

**MIARE**  Minimal Information About an RNAi Experiment. 62

**MIRACLE**  Microarray R-based Analysis of Complex Lysate Experiments. 17, 92, 94–100, 102, 104, 105, 107, 109, 110, 113, 114, 116, 128–131, 133, 134, 156, 183, 184

**miRNA**  microRNA. 6–9, 11, 12, 14, 17, 53, 54, 63, 65, 70–73, 75, 80, 82, 83, 86, 88, 89, 125, 127, 134, 155

**mRNA**  messenger RNA. 1, 7–9, 12, 16
Abbreviations

**MVC** Model View Controller. 108, 109

**NCG** Network of Cancer Genes. 26, 30–33

**NGS** next-generation sequencing. 1, 5, 12, 71, 134


**OOB** out-of-bag error. 23, 175

**PAM** protospacer-associated motif. 10

**PLN** paper laboratory notebook. 45, 46, 123

**pre-miRNA** precursor miRNA. 8, 9

**pri-miRNA** primary miRNA. 8, 9

**QC** quality control. 68, 69

**qRT-PCR** quantitative real-time polymerase chain reaction. 123

**REST** Representational State Transfer. 109, 111, 112, 114, 122, 131

**RISC** RNA induced silencing complex. 8, 9, 71

**RNAi** RNA interference. 7–9, 11, 53, 54, 64, 65, 70, 75, 86–88, 103, 125, 127, 130, 134, 157, 168–170

**RNAice** RNAi and Compound Screen Evaluation. 15, 19, 59, 60, 65, 70, 73, 75–83, 85, 87–89, 116, 117, 125, 127, 128, 130, 134, 155, 156, 183

**ROC** receiver operator characteristics. 25


**SD** standard deviation. 67, 69, 87, 127
Abbreviations

sgRNA  single-guide RNA. 10, 14, 125

shRNA  small hairpin RNA. 14, 125, 134

siRNA  small interfering RNA. 7–9, 14, 51, 54, 67, 73, 75, 82, 83, 86, 88, 125, 155

SSMD  strictly standardized mean difference. 69, 70, 77, 79, 84, 87, 126

TALEN  transcription activator-like effector nuclease. 10

TCGA  The Cancer Genome Atlas. 16, 22, 23, 27, 32, 120, 121

trRNA  trans-activating crRNA. 10

ZFN  zinc-finger nuclease. 10
Appendix A

Supplemental Tables

<table>
<thead>
<tr>
<th>Project Name</th>
<th>Main purpose</th>
<th>Built with</th>
</tr>
</thead>
<tbody>
<tr>
<td>EnzymeTracker</td>
<td>Result &amp; sample management, data analysis</td>
<td>PHP</td>
</tr>
<tr>
<td>SIGLa [115]</td>
<td>Customized experimental work-flows</td>
<td>Java</td>
</tr>
<tr>
<td>iLAP [114]</td>
<td>Data management, analysis, protocol design</td>
<td>Java</td>
</tr>
<tr>
<td>BIKA [200]</td>
<td>Whole lab work-flow for clinical studies</td>
<td>Python</td>
</tr>
<tr>
<td>FreeLIMS [201]</td>
<td>Sample management, reports</td>
<td>Java</td>
</tr>
<tr>
<td>PARPs DB [110]</td>
<td>Protein-protein interaction &amp; data-mining</td>
<td>Perl/Java</td>
</tr>
<tr>
<td>YourLabData [202]</td>
<td>Sample tracking and lab notebook</td>
<td>-</td>
</tr>
<tr>
<td>LabStoRe [203]</td>
<td>Chemical lab stocks</td>
<td>PHP</td>
</tr>
<tr>
<td>LabLog [204]</td>
<td>Project Documentation</td>
<td>Java</td>
</tr>
<tr>
<td>ProteinTracker</td>
<td>Protein production &amp; purification</td>
<td>Java</td>
</tr>
<tr>
<td>AG-LIMS [106]</td>
<td>Genotyping work-flow</td>
<td>Java</td>
</tr>
<tr>
<td>SMS [107]</td>
<td>Gene mutation screening &amp; biobanking</td>
<td>Java</td>
</tr>
<tr>
<td>MMP-LIMS [105]</td>
<td>Genome mapping in maize</td>
<td>Java</td>
</tr>
<tr>
<td>LIPAGE [111]</td>
<td>2D gel electrophoresis based proteomics</td>
<td>PHP</td>
</tr>
<tr>
<td>MicroGen [116]</td>
<td>Microarray information and work-flow</td>
<td>MS-Access</td>
</tr>
<tr>
<td>LISA [112]</td>
<td>Protein crystallography</td>
<td>PHP</td>
</tr>
<tr>
<td>MolabIS [118]</td>
<td>Molecular genetics data</td>
<td>Perl</td>
</tr>
<tr>
<td>SIMBioMS [117]</td>
<td>Linking patient and high-throughput data</td>
<td>Java</td>
</tr>
<tr>
<td>OpenFreezer [113]</td>
<td>Sample management &amp; tracking</td>
<td>PHP/Python</td>
</tr>
<tr>
<td>PiMS [108]</td>
<td>Sample tracking for protein production</td>
<td>Java</td>
</tr>
</tbody>
</table>

Table A.1: A large number of open-source LIMS solutions have already been developed. However, LIMS are often customized towards a very specific type of research data. This is an overview of existing browser-based LIMS solutions, not claiming to be exhaustive.
Table A.2: Depending on its purpose, each LIMS has to fulfill a different set of requirements. Here, we list all those requirements (R1 - R15) that are common for all LIMS where sample management on a large scale is involved. Existing LIMS fulfill these requirements to a varying degree, including OLF. Note: o depicts limited fulfillment. See Supplemental Table A.1 for references.
### Table A.3: Feature comparison between the IARC ELN and OpenLabNotes

Here, we list requirements (R1 - 18) that we identified as important for an ELN.

<table>
<thead>
<tr>
<th>Requirements</th>
<th>IARC ELN</th>
<th>OpenLabNotes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Userstamp and timestamp R1</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Digital signing (author) R2</td>
<td>❌</td>
<td>✓</td>
</tr>
<tr>
<td>Digital cosigning (supervisor) R3</td>
<td>❌</td>
<td>✓</td>
</tr>
<tr>
<td>Notebooks R4</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Projects R5</td>
<td>❌</td>
<td>✓</td>
</tr>
<tr>
<td>Share with specific user R6</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Share with user groups R7</td>
<td>✓</td>
<td>❌</td>
</tr>
<tr>
<td>Comments / annotations R8</td>
<td>✓</td>
<td>❌</td>
</tr>
<tr>
<td>File attachments R9</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Navigation interface R10</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Rich text editor R11</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Editor templates R12</td>
<td>✓</td>
<td>❌</td>
</tr>
<tr>
<td>Sort by title or date R13</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Search by key words R14</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Full text search R15</td>
<td>❌</td>
<td>✓</td>
</tr>
<tr>
<td>Export to PDF R16</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Export to DOCX R17</td>
<td>❌</td>
<td>✓</td>
</tr>
<tr>
<td>Integration with LIMS R18</td>
<td>❌</td>
<td>✓</td>
</tr>
</tbody>
</table>
### Table A.4: Feature comparison between SAVANAH and comparable open source solutions for managing HTS data. The identified requirements are listed here as R1 - R28.

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Mscreen</th>
<th>Screensaver</th>
<th>SAVANAH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open Source</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>User Roles</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Screening libraries</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Track library plate dilutions</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Control wells</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Plate reformatting</td>
<td>✓</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Well volumes</td>
<td>x</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td>Well deprecation</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Active wells</td>
<td>✓</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Multi-layer plate editor</td>
<td>x</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Plate Readouts</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Experiment Meta Data</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Cherry Picking</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td>Plate layout randomization</td>
<td>x</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td>Dose response curves</td>
<td>✓</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>RNAi reagents</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Chemical compounds</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>miRNA reagents</td>
<td>x</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Full text search</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>External databases</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Heatmap</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Scatter plot</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Quality Control</td>
<td>✓</td>
<td>x</td>
<td>✓ (RNAice)</td>
</tr>
<tr>
<td>Data normalization</td>
<td>✓</td>
<td>x</td>
<td>✓ (RNAice)</td>
</tr>
<tr>
<td>Hit discovery</td>
<td>o (SD/MAD)</td>
<td>o</td>
<td>✓ (RNAice)</td>
</tr>
<tr>
<td>Dose response</td>
<td>✓</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Time series</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Array-based readouts</td>
<td>x</td>
<td>x</td>
<td>✓ (MIRACLE)</td>
</tr>
</tbody>
</table>

### Table A.5: Screening libraries can be added to SAVANAH via tab delimited files with a specific header.

<table>
<thead>
<tr>
<th>Plate</th>
<th>Well.position</th>
<th>Catalog.Nr</th>
<th>Probe.ID</th>
<th>Sample.Name</th>
<th>Accession</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A1</td>
<td>410007-00</td>
<td>34143</td>
<td>hsa-mir-384</td>
<td>MIMAT1075</td>
<td>NA</td>
</tr>
<tr>
<td>1</td>
<td>A2</td>
<td>410008-00</td>
<td>34144</td>
<td>hsa-mir-515-5p</td>
<td>MIMAT2826</td>
<td>NA</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>NA</td>
</tr>
</tbody>
</table>
### Table A.6: Feature comparison between RNAice and other open source solutions for analyzing HTS data across requirements previously identified as R1 - R39. Some functionality is provided by external tools such as HTSAnalyzeR or KeyPathwayMiner (KPM). Abbreviations: Negative percentage inhibition (NPI); Standard deviation (SD); Median absolute deviation (MAD); Chemical (Chem.).

<table>
<thead>
<tr>
<th>Requirements</th>
<th>CellHTS2</th>
<th>RNAither</th>
<th>RNAice</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Input</td>
<td>Column Mapping</td>
<td>R1</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Well format conversion</td>
<td>R2</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>PubChem AID</td>
<td>R3</td>
<td>✗</td>
</tr>
<tr>
<td>Quality Control</td>
<td>Visual Inspection</td>
<td>R4</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Z’-factor</td>
<td>R5</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>SSMD</td>
<td>R6</td>
<td>✗</td>
</tr>
<tr>
<td></td>
<td>NPI</td>
<td>R8</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>z-score</td>
<td>R9</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Robust z-score</td>
<td>R10</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Center by mean</td>
<td>R11</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Center by median</td>
<td>R12</td>
<td>✓</td>
</tr>
<tr>
<td>Normalization</td>
<td>Center by shorth</td>
<td>R13</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>B-score</td>
<td>R14</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Bivariate regression</td>
<td>R15</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Loess regression</td>
<td>R16</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Li-Wong rank</td>
<td>R17</td>
<td>✗</td>
</tr>
<tr>
<td></td>
<td>Quantile</td>
<td>R18</td>
<td>✓</td>
</tr>
<tr>
<td>Hit Discovery</td>
<td>+/- k SD</td>
<td>R19</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>+/- k MAD</td>
<td>R20</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Bayes</td>
<td>R21</td>
<td>✗</td>
</tr>
<tr>
<td></td>
<td>SSMD</td>
<td>R22</td>
<td>✗</td>
</tr>
<tr>
<td></td>
<td>Data Export</td>
<td>R23</td>
<td>✓</td>
</tr>
<tr>
<td>Downstream Analysis</td>
<td>Static Plots</td>
<td>R24</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Interactive Plots</td>
<td>R25</td>
<td>✗</td>
</tr>
<tr>
<td></td>
<td>Differential screening</td>
<td>R26</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Consensus hit list</td>
<td>R27</td>
<td>✗</td>
</tr>
<tr>
<td></td>
<td>miRNA families</td>
<td>R28</td>
<td>✗</td>
</tr>
<tr>
<td></td>
<td>miRcancer database</td>
<td>R29</td>
<td>✗</td>
</tr>
<tr>
<td></td>
<td>miRNA targets</td>
<td>R30</td>
<td>✗</td>
</tr>
<tr>
<td></td>
<td>Chem. structure plot</td>
<td>R31</td>
<td>✗</td>
</tr>
<tr>
<td></td>
<td>Similar structures</td>
<td>R32</td>
<td>✗</td>
</tr>
<tr>
<td></td>
<td>Drug targets</td>
<td>R33</td>
<td>✗</td>
</tr>
<tr>
<td></td>
<td>GSOA</td>
<td>R34</td>
<td>HTSAnalyzeR</td>
</tr>
<tr>
<td></td>
<td>GSEA</td>
<td>R35</td>
<td>HTSAnalyzeR</td>
</tr>
<tr>
<td></td>
<td>Network enrichment</td>
<td>R36</td>
<td>HTSAnalyzeR</td>
</tr>
<tr>
<td>Graphical or Web Interface</td>
<td>Data Input</td>
<td>R37</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Result browser</td>
<td>R38</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Reactive Design</td>
<td>R39</td>
<td>✗</td>
</tr>
</tbody>
</table>
### Table A.7: The 19 suspected TNFα modulators that were subjected to secondary screening. The robust z-score obtained by Nickels et al. in comparison to z-score, robust z-score and B-score normalized values in RNAi.

<table>
<thead>
<tr>
<th>Gene Symbol</th>
<th>robust z-score (Nickels et al.)</th>
<th>z-score</th>
<th>robust z-score</th>
<th>B-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>CKB</td>
<td>-4.24</td>
<td>-2.13</td>
<td>-2.55</td>
<td>-2.44</td>
</tr>
<tr>
<td>CCL25</td>
<td>-3.24</td>
<td>-1.44</td>
<td>-1.64</td>
<td>-1.98</td>
</tr>
<tr>
<td>CKAP5</td>
<td>-3.79</td>
<td>-3.06</td>
<td>-3.87</td>
<td>-3.73</td>
</tr>
<tr>
<td>KIAA0999</td>
<td>-3.44</td>
<td>-1.39</td>
<td>-1.49</td>
<td>-0.64</td>
</tr>
<tr>
<td>NOP17</td>
<td>-3.13</td>
<td>-1.22</td>
<td>-1.35</td>
<td>-1.10</td>
</tr>
<tr>
<td>SEZ6L2</td>
<td>-2.95</td>
<td>-1.73</td>
<td>-1.86</td>
<td>-0.85</td>
</tr>
<tr>
<td>GPR119</td>
<td>-2.78</td>
<td>-1.90</td>
<td>-2.27</td>
<td>-2.00</td>
</tr>
<tr>
<td>KIF2B</td>
<td>-2.75</td>
<td>-1.72</td>
<td>-1.96</td>
<td>-1.47</td>
</tr>
<tr>
<td>MS4A4A</td>
<td>-2.93</td>
<td>-1.71</td>
<td>-2.11</td>
<td>-1.25</td>
</tr>
<tr>
<td>CLTC</td>
<td>-2.74</td>
<td>-1.56</td>
<td>-1.88</td>
<td>-1.38</td>
</tr>
<tr>
<td>ITGB4BP</td>
<td>-2.67</td>
<td>-1.61</td>
<td>-2.18</td>
<td>-2.06</td>
</tr>
<tr>
<td>CASP4</td>
<td>-2.56</td>
<td>-1.39</td>
<td>-1.59</td>
<td>-1.42</td>
</tr>
<tr>
<td>GH2</td>
<td>-2.57</td>
<td>-1.48</td>
<td>-1.59</td>
<td>-0.03</td>
</tr>
<tr>
<td>MEN1</td>
<td>-2.55</td>
<td>-1.37</td>
<td>-1.56</td>
<td>-1.46</td>
</tr>
<tr>
<td>C10orf664</td>
<td>-3.11</td>
<td>-2.01</td>
<td>-2.35</td>
<td>-1.82</td>
</tr>
<tr>
<td>PRO1855</td>
<td>-3.04</td>
<td>-2.14</td>
<td>-2.65</td>
<td>-2.30</td>
</tr>
<tr>
<td>FLJ10980</td>
<td>-2.63</td>
<td>-1.83</td>
<td>-2.29</td>
<td>-1.13</td>
</tr>
<tr>
<td>PTK9</td>
<td>-2.89</td>
<td>-1.76</td>
<td>-2.13</td>
<td>-1.49</td>
</tr>
<tr>
<td>ROBO4</td>
<td>-2.56</td>
<td>-1.27</td>
<td>-1.45</td>
<td>-0.74</td>
</tr>
</tbody>
</table>

### Table A.8: GO molecular function terms found to be significantly overrepresented or enriched in hits found in RNAi screening for TNFα modulators. Only those results are shown, where the adjusted p-value (Benjamini-Hochberg) was significant in both gene set overrepresentation analysis (GSOA) and gene set enrichment analysis (GSEA). The minimal gene set size was 10 and the number of permutations for GSEA was 1000.

<table>
<thead>
<tr>
<th>GO id</th>
<th>Gene Set Term</th>
<th>GSOA p adj</th>
<th>GSEA p adj</th>
</tr>
</thead>
<tbody>
<tr>
<td>GO:0003899</td>
<td>DNA-directed RNA polymerase activity</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>GO:0003735</td>
<td>structural constituent of ribosome</td>
<td>1.88e-08</td>
<td>0</td>
</tr>
<tr>
<td>GO:0044822</td>
<td>poly(A) RNA binding</td>
<td>1.17e-06</td>
<td>0</td>
</tr>
<tr>
<td>GO:0019843</td>
<td>rRNA binding</td>
<td>0.01</td>
<td>0.05</td>
</tr>
</tbody>
</table>
### Table A.9: GO cellular compartment terms found to be significantly overrepresented or enriched in hits found in RNAi screening for TNFα modulators. Only those results are shown, where the adjusted p-value (Benjamini-Hochberg) was significant in both gene set overrepresentation analysis (GSOA) and gene set enrichment analysis (GSEA). The minimal gene set size was 10 and the number of permutations for GSEA was 1000.

<table>
<thead>
<tr>
<th>GO id</th>
<th>Gene Set Term</th>
<th>GSOA p adj.</th>
<th>GSEA p adj.</th>
</tr>
</thead>
<tbody>
<tr>
<td>GO:0022627</td>
<td>cytosolic small ribosomal subunit</td>
<td>9.11e-14</td>
<td>0</td>
</tr>
<tr>
<td>GO:0005665</td>
<td>DNA-directed RNA polymerase II, core complex</td>
<td>6.96e-08</td>
<td>0</td>
</tr>
<tr>
<td>GO:0015935</td>
<td>small ribosomal subunit</td>
<td>1.88e-06</td>
<td>0</td>
</tr>
<tr>
<td>GO:0005730</td>
<td>nucleolus</td>
<td>5.83e-05</td>
<td>0</td>
</tr>
<tr>
<td>GO:0005829</td>
<td>cytosol</td>
<td>0.028</td>
<td>0</td>
</tr>
<tr>
<td>GO:0005840</td>
<td>ribosome</td>
<td>0.01</td>
<td>0</td>
</tr>
</tbody>
</table>

### Table A.10: GO biological process terms found to be significantly overrepresented or enriched in hits found in RNAi screening for TNFα modulators. Only those results are shown, where the adjusted p-value (Benjamini-Hochberg) was significant in both gene set overrepresentation analysis (GSOA) and gene set enrichment analysis (GSEA). The minimal gene set size was 10 and the number of permutations for GSEA was 1000.

<table>
<thead>
<tr>
<th>GO id</th>
<th>Gene Set Term</th>
<th>GSOA p adj.</th>
<th>GSEA p adj.</th>
</tr>
</thead>
<tbody>
<tr>
<td>GO:0016032</td>
<td>viral process</td>
<td>1.77e-13</td>
<td>0</td>
</tr>
<tr>
<td>GO:000184</td>
<td>nuclear-transcribed mRNA catabolic</td>
<td>1.44e-09</td>
<td>0</td>
</tr>
<tr>
<td>GO:0019083</td>
<td>viral transcription</td>
<td>1.53e-11</td>
<td>0</td>
</tr>
<tr>
<td>GO:006413</td>
<td>translational initiation</td>
<td>1.53e-11</td>
<td>0</td>
</tr>
<tr>
<td>GO:006415</td>
<td>translational termination</td>
<td>2.80e-11</td>
<td>0</td>
</tr>
<tr>
<td>GO:0010467</td>
<td>gene expression</td>
<td>3.29e-11</td>
<td>0</td>
</tr>
<tr>
<td>GO:006414</td>
<td>translational elongation</td>
<td>1.45e-10</td>
<td>0</td>
</tr>
<tr>
<td>GO:0019058</td>
<td>viral life cycle</td>
<td>8.35e-10</td>
<td>0</td>
</tr>
<tr>
<td>GO:006614</td>
<td>SRP-dependent cotranslational protein targeting to membrane</td>
<td>9.10e-10</td>
<td>0</td>
</tr>
<tr>
<td>GO:0016071</td>
<td>mRNA metabolic process</td>
<td>2.78e-09</td>
<td>0</td>
</tr>
<tr>
<td>GO:0016070</td>
<td>RNA metabolic process</td>
<td>1.15e-08</td>
<td>0</td>
</tr>
<tr>
<td>GO:006412</td>
<td>translation</td>
<td>3.27e-07</td>
<td>0</td>
</tr>
<tr>
<td>GO:006370</td>
<td>7-methylguanosine mRNA capping</td>
<td>6.20e-07</td>
<td>0</td>
</tr>
<tr>
<td>GO:0050434</td>
<td>positive regulation of viral transcription</td>
<td>4.23e-07</td>
<td>0</td>
</tr>
<tr>
<td>GO:006368</td>
<td>transcription elongation from RNA polymerase II promoter</td>
<td>1.94e-05</td>
<td>0</td>
</tr>
<tr>
<td>GO:0044267</td>
<td>cellular protein metabolic process</td>
<td>3.79e-05</td>
<td>0</td>
</tr>
<tr>
<td>GO:006367</td>
<td>transcription initiation from RNA polymerase II promoter</td>
<td>0.01</td>
<td>0</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>KEGG id</th>
<th>Gene Set Term</th>
<th>GSOA p adj.</th>
<th>GSEA p adj.</th>
</tr>
</thead>
<tbody>
<tr>
<td>hsa03010</td>
<td>Ribosome</td>
<td>2.58e-11</td>
<td>0</td>
</tr>
<tr>
<td>hsa03020</td>
<td>RNA polymerase</td>
<td>8.24e-06</td>
<td>0</td>
</tr>
</tbody>
</table>

Table A.11: KEGG pathways found to be significantly overrepresented or enriched in hits found in RNAi screening for TNFα modulators. Only those results are shown, where the adjusted p-value (Benjamini-Hochberg) was significant in both gene set overrepresentation analysis (GSOA) and gene set enrichment analysis (GSEA). The minimal gene set size was 10 and the number of permutations for GSEA was 1000.

<table>
<thead>
<tr>
<th>Requirements</th>
<th>SuperCurve</th>
<th>RPPanalyzer</th>
<th>RIMS</th>
<th>MIRACLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample Management</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Plate Layouts R3</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Slide Layouts R4</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Slide copies R6</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Multiple readouts R7</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Plate Readouts R8</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Virtual Spotting R9</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Data Analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Visualization R10</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Surface adjustment R11</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Quantification R12</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Normalization R13</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Significance R14</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Correlation R15</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Timecourse R16</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Network Analysis R17</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>x</td>
</tr>
</tbody>
</table>

Table A.12: Features of MIRACLE and other open source solutions for processing RPPA data. Typical requirements in the field of RPPA sample management and data analysis are listed as R1 - R17.
Appendix B

Supplemental Figures
Figure B.1: All 53 selected features of the gene expression model. Taken from [80].
Figure B.2: All 38 selected features of the methylation model. Taken from [80].
Figure B.3: The top 50 features of the combined model that have a mean decrease in gini index > 2. Taken from [80].
Figure B.4: The .632 bootstrap error over 10 iterations (left), the average AUC (AAUC) (middle), and the distribution of the number of variables across bootstrap iterations (right) after applying random forests to the different models. Here, samples labeled as normal have been excluded from model creation. For comparison, the AAUC of the control model is also shown. Taken from [80].

Figure B.5: The classification and out-of-bag error (OOB) of each subtype for the control model (A), the gene expression model (B), the methylation model (C) and the combined model (D) excluding samples labeled by PAM50 as normal and depending on the number of trees already created in the feature selection process. Taken from [80].
Figure B.6: Example of a heatmap covering all plates in a screen. The plot is divided into facets, where each facet corresponds to a particular plate. The color gradient indicates the normalized signal value. Black arrows indicate samples that are considered as hits with the selected criteria.
Figure B.7: A: The raw data of the differential screen for TNFα. Channel 1 depicts a general readout of cell viability, while channel 2 corresponds to an indirect reporter of TNFα activity. The signal density plots of the two different replicates do not agree and are skewed towards promoting activity in both channels. B: A z-score normalization yields a better congruence of the density plots. C: The robust z-score achieves a nearly perfect congruence of the density plots, which appear less skewed. D: The B-score also achieves nearly perfect congruence and eliminates the skewing problem. However, the density plots do not appear to follow the Gaussian distribution any more.
Figure B.8: A: Several microtiter plates can serve as source for a single RPPA slide. B: Each plate is subjected to multiple extractions, depending on the size of the extraction head. C: Each extraction is deposited on the slide and the extraction head is moved slightly to place samples next to each other. This gives rise to as many sample blocks as there are pins on the extraction head. D: Samples are previously diluted to allow measurement in a dynamic range. In addition, samples are spotted with varying number of depositions.

Figure B.9: To cover a broad dynamic range, samples are typically spotted in a dilution series on an RPPA. Subsequently, the individual spots have to be merged into a single representative value in a process called quantification. Taken from [176].
Figure B.10: In some cases, a block shift is introduced during spot recognition of RPPAs (A). As a consequence, a false signal intensity is associated with a given sample. MIRACLE can correct this by shifting the signal intensities to their intended position (B). Taken from [176].
Figure B.11: Surface normalization in Rmiracle using positive control spots. A: RPPA slide with quality problems shows uneven antibody staining. B: Positive control spots are included at the beginning and the end of each block. C: These can be used to calculate a generalized additive model reflecting a smoothing surface. From this, an individual correction factor can be calculated for and applied to each single spot.
Appendix C

Availability

C.1 Version Control System and Public Code Repository

Git was chosen for managing the source code produced during this thesis. This allows storing source code in github \(^1\), which is currently developing into the de facto standard for open source projects in general and in the scientific community in particular. All source code described in this thesis is available through the github accounts NanoCAN and mlist. More detailed URLs (last access 06/05/2015) follow below.

C.2 Open Source License

The Gnu Public License v3 (GPLv3) was chosen as open-source license for the code produced in this thesis. It is a common choice for non-commercial open-source code that can be freely distributed, altered and re-used as long as the original source is referenced. The GPLv3 further prohibits the commercial use of this source code by any entity.

\(^1\)http://www.github.com (accessed 06/04/2015)
Appendix C. Code Repositories and Demo Applications

C.3 Classification of Breast Cancer Subtypes By Combining Gene Expression and DNA Methylation Data

The source code used to download and process the input data, to apply random forest with recursive feature elimination (using the varSelRF R package) and parallelization, as well as all the code used to produce the figures and tables for Chapter 2 is available at

https://github.com/mlist/IB2014

C.4 OpenLabFramework

The source code for OLF is divided into several repositories reflecting the modular nature of the program.

- OpenLabBackend: https://github.com/NanoCAN/OpenLabBackend
- OpenLabGeneTracker: https://github.com/NanoCAN/OpenLabGeneTracker
- OpenLabStorage: https://github.com/NanoCAN/OpenLabStorage
- OpenLabBarcode: https://github.com/NanoCAN/OpenLabBarcode
- OpenLabAttachments: https://github.com/NanoCAN/OpenLabAttachments

A live demo of OLF can be accessed at http://www.nanocan.dk/openlabframework/demo (username: admin, password: demo0815).

Further documentation can be found at https://github.com/NanoCAN/OpenLabFramework/wiki

C.5 OpenLabNotes

The original implementation of OpenLabNotes was created by Michael Franz during an individual study activity. The source code can be found at:
https://github.com/michaelfranz89/OpenLabNotes

The source code was subsequently extended to provide additional features, such as detached signatures or notebooks, and to better integrate with existing OLF functionality, such as linking to samples, projects, and uploaded files. The source code described in this thesis can be found at:

https://github.com/NanoCAN/OpenLabNotes

OpenLabNotes is part of OLF and uses the same documentation and demo installation as described above.

C.6 HTSbackend

As described in Chapter 8, SAVANAH and MIRACLE share common functionality, i.e. they both operate with sample information with regards to microtiter plates. To avoid code redundancy and to increase maintainability, this common functionality was factored into a separate code repository that can be integrated as a Grails plug-in. This source code can be found at:

https://github.com/NanoCAN/HTSbackend

C.7 SAVANAH

The source code for SAVANAH will be available at this URL after its public release:

https://github.com/NanoCAN/SAVANAH (username: demo, password: demo0815)

A live demo that can interact with the live demo of OLF is available at:

http://www.nanocan.org/savanah/demo

A detailed user manual is in preparation.

C.8 RNAice

The source code for RNAice will be available at this URL after its public release:
Appendix C. Code Repositories and Demo Applications

https://github.com/NanoCAN/RNAice

A demo installation is hosted at:

http://www.nanocan.org/rnaice/demo

A detailed user manual is in preparation.

C.9 MIRACLE

The source code for MIRACLE can be found at:

https://github.com/NanoCAN/MIRACLE

The code for Rmiracle, the R shiny application that integrates with MIRACLE and allows processing and analysis of RPPA data, is found at:

https://github.com/NanoCAN/Rmiracle

MIRACLE allows creating zoomable views of slides using the OpenSeaDragon javascript library. To optimally utilize this functionality, a Grails plug-in was developed and contributed to the community. The source code can be found at:

https://github.com/mlist/grails-openseadragon-plugin

The plug-in is listed at the Grails project page:

http://grails.org/plugin/open-seadragon

A user manual for MIRACLE can be found at the project page:

http://nanocan.github.io/MIRACLE/

A live demo application that interacts with the SAVANAH and OLF demo applications can be accessed at:

http://www.nanocan.org/miracle/demo (username: demo, password: demo0815)
Appendix D

Supplemental Methods

D.1 Details on Bayesian Hit Detection

Zhang et al. have proposed two different Bayesian models for hit discovery based on the concept of borrowing strength across the entire experiment (Chapter 6.3.3.2). Model 1 relies only on negative control wells, while model 2 relies on negative as well as positive control wells. Since model 2 has more parameters than model 1 and since it is, according to Zhang et al. outperformed by model 1, we focus solely on the description of model 1.

Likelihood:

Let $X_1, X_2, \ldots, X_k$ be log transformed signal values. For calculating the likelihood, we assume that each sample well follows a standard distribution

$$X_i | \mu_i \sim N(\mu_i, \sigma^2)$$  \hspace{1cm} (D.1)

where $\mu_i$ denotes the center of the distribution of sample $i$ and $\sigma^2$ denotes the variance.

The following hypotheses can be tested:
\[ H_0: \text{sample } i \in \{ \text{no effect} \} \]  
\[ H_1: \text{sample } i \in \{ \text{activation effect} \} \]  
\[ H_2: \text{sample } i \in \{ \text{inhibitor effect} \} \]

Prior:

The prior distribution is defined as

\[ \mu_i \sim N(\theta_0, \sigma^2) \]  \hspace{1cm} (D.5)

where \( \theta_0 \) denotes the center and \( \sigma^2 \) denotes the variance of the negative control. For the negative control wells, we assume independence and equal distribution

\[ X_i | \mu_N \sim N(\mu_N, \sigma^2) \]  \hspace{1cm} (D.6)

where \( \mu_N \) is the mean of the negative control. The variability of the negative control can be assumed to be identical across plates and is thus pooled.

For each plate, the negative control wells are denoted as \( Y_{ij} \), where \( i \) denotes the index of the negative control and \( j \) denotes the plate number. Each negative control \( Y_{ij} \) deviates by \( d_{ij} \) from the plate-wise mean of the negative controls denoted as \( Y_{\bullet j} \)

\[ d_{ij} = Y_{ij} - Y_{\bullet j} \]  \hspace{1cm} (D.7)

The estimate of the variance \( \hat{\sigma}^2 \) of the negative control is calculated as
Appendix D. Supplemental Methods

\[ \hat{\sigma}^2 = \frac{\sum_i \sum_j d_{ij}^2}{\sum_j n_j - A} \]  

(D.8)

Assuming that samples \( X_i \) have the same variance in a given plate, we can calculate \( \text{Var}(X_i) \) based on the variance of the negative control plus additional variance estimated from the sample itself, denoted as \( \text{Var}(X_i) \).

\[ \text{Var}(X_i) = \hat{\sigma}^2 + \tau^2 \]  

(D.9)

\[ \tau^2 = \max \left[ \text{Var}(X_i) - \hat{\sigma}^2, 0 \right] \]  

(D.10)

**Posterior**: The prior mean of sample \( i \) is \( Y_{w} \), where \( w \) denotes the plate index of \( i \). The prior variance is denoted as \( \tau^2 \). The posterior distribution is then

\[ \mu_i|X_i \sim N \left( \frac{\sigma^2 \theta_0 + \tau^2 X_i}{\sigma^2 + \tau^2}, \frac{\sigma^2 \tau^2}{\sigma^2 + \tau^2} \right) \]  

(D.11)

\[ \sim N \left( \frac{\sigma^2 Y_{w} + \tau^2 X_i}{\text{Var}(X_i)}, \frac{\sigma^2 \tau^2}{\text{Var}(X_i)} \right) \]  

(D.12)

where \( \theta_0 \) is given by \( Y_{w} \).

**Bayesian hypothesis testing**: The Bayesian framework allows to detect hits based on the signal \( X_i \) of a sample and the center of the distribution of the negative reference \( \theta_0 \) and is weighted by the variability of both, the sample \( \text{Var}(X_i) \) and the negative control \( \tau \). The p-values under the different hypotheses introduced above are calculated as follows:

\[ P(H_1|X_i) = P(\mu_i - \theta_0 > \alpha | X_i) \]  

(D.13)

\[ P(H_2|X_i) = P(\mu_i - \theta_0 < -\alpha | X_i) \]  

(D.14)

\[ P(H_0|X_i) = P(|\mu_i - \theta_0| \leq \alpha | X_i) \]  

\[ = 1 - P(H_1|X_i) - P(H_2|X_i) \]  

(D.15)
Based on these p-values, the $FDR$ can be calculated and controlled via multiple testing correction using, for instance, the Benjamini-Hochberg method [142].

### D.2 Authentication to Grails Applications in R

```r
authenticate.to.grails <- function(baseUrl="http://localhost:8080/MIRACLE/", user, password, verbose=F)
{
  require(RCurl)
  loginUrl = paste(baseUrl, "login/auth", sep="")
  authenticateUrl = paste(baseUrl, "j_spring_security_check", sep="")
  cat(paste("trying to authenticate user", user))
  agent="Mozilla/5.0"

  #Set RCurl parameters
  curl = getCurlHandle()
  curlSetOpt(ssl.verifypeer=FALSE, timeout=60, cookiefile=tempfile(),
             cookiejar=tempfile(), useragent=agent, followlocation = TRUE, curl=
             curl, verbose=verbose)

  #Open login page
  getURL(loginUrl, curl=curl)

  #Post login form
  postForm(authenticateUrl, .params=list(j_username=user, j_password=
                                           password), curl=curl, style="POST")

  return(curl)
}
```

Listing D.1: R method to authenticate to a Grails web application