Analysis of High Content RNA Interference Screens at Single Cell Level

Master's Thesis
Written by
Nicolas Bennett

Supervised by
Dr. Anna Drewek
and
Prof. Dr. Peter Bühlmann

Seminar for Statistics
Swiss Federal Institute of Technology
CH-8092 Zürich

Received August 25th, 2015

Little Strokes,
Fell great Oaks

Poor Richard's
Almanack, 1750
Benjamin Franklin

Preface

This thesis is submitted in partial fulfillment of the requirements for a Master's Degree in Interdisciplinary Sciences with a Major in Applied Mathematics and Computational Biology at the Swiss Federal Institute of Technology (Eidgenössische Technische Hochschule, ETH) Zürich. It contains work performed from April to August 2015 under the supervision of Dr. Anna Drewek and Prof. Dr. Peter Bühlmann.

All investigated data originates from the large-scale RNA interference (RNAi) screening experiment InfectX (2010–2014) and the present work adds to its follow-up venture TargetInfectX (ongoing). Both Research, Technology and Development (RTD) projects are funded by SystemsX, the Swiss Initiative in Systems Biology, and set out to comprehensively study the human infectome for a set of common bacterial and viral pathogens. While InfectX was more focused on developing unified wet-lab procedures yielding comparable results over a broad range of pathogens and establishing protocols for microscopy and image analysis, TargetInfectX builds on the established datasets and places an emphasis on exploration of phenotypic space.

Picking up at the introductory quote by Benjamin Franklin, exhaustively identifying the human protein network underlying infection is an enormous task, in light of which even projects such as the InfectX program play only a fractional role towards the overall objective. It is my hope that this work may contribute an ever so small 'stroke' towards better understanding the data collected by InfectX, which in turn is only a small part of felling the tree of elucidating pathogenicity in human cells.

The cover page is a graphical representation of single cell feature data as investigated. It shows the MTOR well (H6) of plate J107-2C; circle size corresponds

to cell area, coloring is based on mean intensity of the actin channel measured throughout entire cell bodies and filled circles are drawn for cells considered infected while outlined circles indicate healthy cells.

Declaration of Originality

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisors. None of this material has been submitted in any form for another degree or diploma at any institute of tertiary education other than the Swiss Federal Institute of Technology Zürich. The results, figures, tables and text are original, except otherwise indicated by reference to the respective authors and/or organizations.

Furthermore, I confirm that I have committed none of the forms of plagiarism as described in the "Citation Etiquette" information sheet by ETH Zürich and have cited all my sources fully and verifiably in the bibliography. I have documented all methods, data and processes truthfully and I have not manipulated any data. I am aware that the work may be screened electronically for plagiarism.

Acknowledgments

Several people were essential to the realization of this project, some of whom I wish to mention explicitly. Firstly, I would like to express my gratitude to Anna Drewek for her support and guidance throughout all stages of this project. I enjoyed the frequent meetings and benefited greatly from her insightful remarks. It was a pleasure working with her and learning from her. Along with her, I would like to thank Peter Bühlmann for having me be a part of his research group at the Seminar for Statistics. I am much obliged for this opportunity.

Of course, none of this would have been possible without the work put forward by the InfectX/TargetInfectX consortia. I am grateful to all who made possible this incredibly large, detailed and valuable dataset; the biologists who worked out wet-lab procedures, the machine learning experts responsible for image analysis, the modelers dealing with data normalization and all support staff, responsible for securing funding, providing computational resources and handling the administrative overhead of such a large-scale initiative. I would like to give special thanks to Mario Emmenlauer for his assistance in dealing with many issues surrounding data access and formatting.

Finally, I would like to thank both Judith Wyss an my family for all their love and support, especially during this stressful time.

Abstract

Infectious diseases are among the leading causes of death worldwide and the evolution of antimicrobial resistance poses a troubling development in cases where our only effective line of defense is based on distribution of antibiotic agents. One possible way out of this perilous situation comes by the alternative approach of host directed therapeutics, which in turn warrants the meticulous study of the human infectome. Therefore, large-scale studies such as genome-wide siRNA knockdown experiments as performed by the InfectX/slash TargetInfectX consortia are of great importance.

The richness of datasets resulting from image-based high throughput RNAi screens permits a broad range of possible analysis approaches to be employed. The present study investigates cellular phenotypes as induced by gene knockdown, with a focus on the effect of pathogen infection by applying generalized linear models (GLMs) to single cell measurements. In order to simplify handling of such datasets, an R package is presented that is capable of fetching queried data from a centralized data store and producing a data structure capable of efficiently representing the logic of an assay plate. Convenience functions to preprocess, manipulate and normalize the resulting objects are provided, as is a caching system that helps to significantly speed up common operations.

GLM analysis of phenotypic response from knockdown and infection was attempted, but did not yield satisfactory results, most probably due to issues with data normalization. In order to facilitate the simultaneous study of measurements originating from multiple assay plates, several normalization schemes were explored, including Z- and B-scoring, as well as modeling technical artifacts with multivariate adaptive regression splines (MARS). While some improvements of data quality were observed, experimental sources of error could not be sufficiently controlled for meaningful GLM regression.

Contents

List of Figures

List of Tables

Code Listings

Abbreviations and Acronyms

AAV adeno-associated virus
ABM actin based motility
ActA *Listeria* actin assembly-inducing protein
ADP adenosine diphosphate
Ago Argonaute protein family
AIC Akaike information criterion
AIDS acquired immune deficiency syndrome
AMR antimicrobial resistance
ANOVA analysis of variance
API application programming interface
Arp2/3 actin-related-protein 2 and 3
AS application server
ATCC American Type Culture Collection
ATP adenosine triphosphate
AvrA *Salmonella* avirulence protein A
Bep *Bartonella* effector protein
BrCV *Brucella*-containing vacuole
BSA bovine serum albumin protein
cAMP cyclic adenosine monophosphate
CAR coxsackievirus and adenovirus receptor
CBA Columbia base agar
CCD charge-coupled device
CCP clathrin-coated pit
CCV clathrin-coated vesicles
Cdc42 cell division control protein 42 homolog
CFL1 cofilin-1
CLTC clathrin heavy chain 1
CPU central processing unit
CRAN Comprehensive R Archive Network
CSD cat scratch disease
Cy cyanine dye
DAPI 4′,6-diamidino-2-phenylindole
DGCR8 DiGeorge syndrome critical region gene 8
DMEM Dulbecco modified Eagle medium
DNA deoxyribonucleic acid
DNM2 dynamin-2
dsDNA double stranded DNA
dsRNA double stranded RNA
DSS data store server
DTIS decision tree infection scoring
EHEC enterohemorrhagic *E. coli*
EPEC enteropathogenic *E. coli*

ER endoplasmic reticulum
EV extracellular virion
FBS fetal bovine serum
FDR false discovery rate
GalNAc N-acetylgalactosamine
GAP GTPase activating protein
GCV generalized cross validation
GDP guanosine diphosphate
GEF guanine nucleotide exchange factor
GFP green fluorescent protein
GLM generalized linear model
GNI gross national income
GTP guanosine triphosphate
HDT host directed therapeutics
Hil *Salmonella* hyperinvasion locus
HIV human immunodeficiency virus
HPC high performance computing
HSP heat shock protein
HTS high throughput screening
I/O input/output
ICAM-1 intercellular adhesion molecule 1
Ics *Shigella* intracellular spread protein
IL-8 interleukin-8
Inl *Listeria* internalin
Inv *Salmonella* invasion protein
Ipa *Shigella* invasion protein
Ipg *Shigella* invasion plasmid gene
IRLS iteratively reweighted least squares
Kif11 kinesin family member 11
LB lysogeny broth
LCV *Legionella*-containing vacuole
LED light-emitting diode
LIL list of lists
LLO listeriolysin O
LODER localized, sustained siRNA delivery technology
LOF loss-of-function
LPS lipopolysaccharide
M cells microfold cells
MAD median absolute deviation
MAPK mitogen-activated protein kinase
MARS multivariate adaptive regression splines
MCC Matthews correlation coefficient
miRNA micro RNA
MLE maximum likelihood estimator
MOI multiplicity of infection
mRNA messenger RNA
MTOR mechanistic target of rapamycin
MV mature virion
Mxi *Shigella* membrane expression of Ipa
N-WASP neural WASP
NF-κB nuclear factor κ-light-chain-enhancer of activated B cells
NPF nucleation promoting factor
OLS ordinary least squares
Omp *Salmonella* outer-membrane protein
OOP object oriented programming
Org *Salmonella* oxygen-regulated gene
Osp outer *Shigella* protein
OTE off-target effects
PACT protein activator of protein kinase PKR
PBS phosphate-buffered saline
PC principal component
PCA principal component analysis
PFA paraformaldehyde
Pho phosphate metabolism protein
PI(4)P phosphatidylinositol 4-phosphate
PI(4,5)P_2 phosphatidylinositol 4,5-bisphosphate
PI(5)P phosphatidylinositol 5-phosphate
PIK3R3 phosphoinositide-3-kinase regulatory subunit 3

PKR protein kinase RNA-activated
PMM parallel mixed model
Prg *Salmonella* PhoP-repressed gene
pri-miRNA primary miRNA
Rab Ras-related in brain
Rac1 Ras-related C3 botulinum toxin substrate 1
Ran Ras-related nuclear protein
Ras rat sarcoma family protein
RC reference class
RFP red fluorescent protein
RhoG Ras homology growth-related
RIPK4 receptor-interacting serine/threonine kinase 4
RISC RNA-induced silencing complex
RITS RNA-induced transcriptional silencing complex
RNA ribonucleic acid
RNAa RNA activation
RNAi RNA interference
RNase ribonuclease
rRNA ribosomal RNA
RT reverse transcriptase
rt room temperature
RTD Research, Technology and Development
SCV *Salmonella*-containing vacuole
shRNA short hairpin RNA
Sic *Salmonella* invasion chaperone protein
Sif *Salmonella*-induced filament
Sip *Salmonella* invasion protein
siRNA short interfering RNA
Sop *Salmonella* outer protein
Spa *Salmonella* surface presentation of antigen
SPI *Salmonella* pathogenicity island
spp. *species pluralis*; multiple, not explicitly enumerated species
SptP *Salmonella* protein tyrosine phosphatase
Spv *Salmonella* plasmid virulence
Ssa *Salmonella* secretion system apparatus protein
Ssc *Salmonella* secreted chaperone protein
ssDNA single stranded DNA
Sse *Salmonella* secreted effector protein
Ssp *Salmonella*-secreted protein
ssRNA single stranded RNA
subsp. subspecies
SVM support vector machine
T3SS type III secretion system
T4SS type IV secretion system
TAP tapasin
TAR trans-activation response element
TGFBR1 transforming growth factor, beta receptor 1
Tir translocated intimin receptor
TLN1 talin-1
TLR toll-like receptor
TRBP TAR RNA-binding protein
VEGF vascular endothelial growth factor
Vir virulence spread protein
WASP Wiskott-Aldrich syndrome protein
WHO World Health Organization

Chapter 1

Introduction

Infectious diseases have played an undeniably important role in human history. With human populations becoming sufficiently aggregated to sustain direct life cycle viral and bacterial infectants around 2000 BC, devastating invasions of a growing number of pathogens started to occur (Dobson and Carper 1996). One of the earliest well documented incidence of a large-scale epidemic is known as the Plague of Athens. Starting in 430 BC and lasting roughly three years, a highly infectious disease killed 75'000 to 100'000 people or 25% of Athen's population. This catastrophic event is attributed either to smallpox, a viral infection with *Variola major*, or typhus, caused by *Rickettsia* bacteria (Littman 2009).

The bacterium *Yersinia pestis* is responsible for three major plague pandemics in the early and late middle ages, as well as in the late 19th century. Originating in northern Africa in 523 AD and spreading around the Mediterranean basin throughout the years 541–546, the Plague of Justinian is assumed to have killed up to half of the population of affected areas. The effect on cities was disproportionately severe. In Constantinople, for example, an estimated 230'000 people out of 375'000 lost their lives to the disease (Treadgold 1997).

Returning in the years 1347–1351, known today as the Black Death, a plague pandemic again wiped out around half of Europe's population. Death toll estimates range from 15 to 23.5 million (Zietz and Dunkelberg 2004). Leaving behind a grim cultural heritage, this catastrophe had a lasting effect on economic and social structures in Europe. The third large-scale outbreak started around 1855 in southern China and quickly spread to Japan, Taiwan and India again wreaking havoc on the affected population. Facilitated by the advent of ocean-spanning trade, the bubonic plague this time reached many inhabited

areas, including South and Central America, the United States, South Africa and Australia.

The import of diseases such as smallpox, measles (an infection with the Measles virus) and typhus to the Americas during the European invasion of the New World had grave repercussions for the indigenous population, carrying no natural resistance towards the newly introduced pathogens. It is estimated that the population of present-day Mexico fell from 20 million to 1.6 million over the course of the 16th century due to multiple disease epidemics, critically contributing to the successful colonization of the new continents (Dobson and Carper 1996).

Cholera and influenza are further contagious diseases with high mortality rates, responsible for global epidemics. *Vibrio cholerae*, a bacterium which infects the intestine, became widespread in the early 19th century and led to seven pandemics since, the last of which only started in 1961. Antibacterial treatment of sewage and purification of drinking water greatly help to prevent and contain spreading of the disease but in areas with inadequate sanitation, such as Haiti after the 2010 earthquake, it remains a pathogen difficult to control. The influenza virus causes seasonal epidemics characterized by low lethality rates among people with intact immune systems[1]. Irregularly occurring influenza pandemics, initiated by zoonosis of new virus strains, against which no natural immunity exists, however, are accompanied by much higher lethality rates. The most significant such event is known today as the Spanish flu pandemic of 1918, costing the lives of 50–100 million, nearly half of which were young, healthy adults (Taubenberger and Morens 2006).

In addition to diseases plaguing humanity for centuries, new ones continually emerge. Human immunodeficiency virus (HIV) is believed to have transferred from non-human primates in the early 20th century and the recent outbreaks of severe acute respiratory syndrome and swine flu serve as reminders of such occurrences.

Despite development of means to treat and prevent many diseases, infectious pathogens remain a serious threat to global health. In 2012, an estimated total of 58.3 million people died (20.1% in high, 29.4% in upper-middle, 36.5% in lower-middle and 14% in low income countries). Figure 1.1 partitions the total death count into World Bank income groups and causes. In low income countries, infective diseases are the most prevalent cause of death (39.6%), followed by maternal and perinatal complications with substantial margin (20.8%). In

[1] In spite of low lethality, these seasonal epidemics still incur significant economic damages. The World Health Organization estimates annual health care costs and loss of productivity due to influenza at US $71–176 billion for the United States of America alone (World Health Organization 2003).

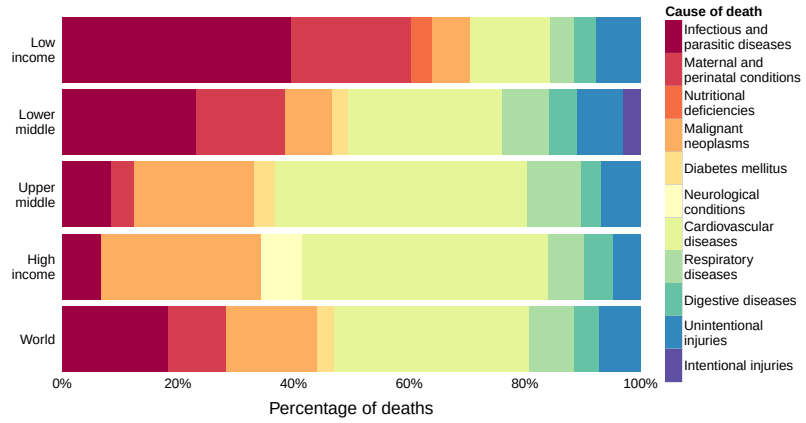


Figure 1.1: Relative frequencies of death causes in 2012 by World Bank income groups. Binning is based on gross national income (GNI) per capita and the thresholds are $1045 or less for low income, $1046 to $4125 for lower-middle, $4126 to $12745 for upper-middle and $12746 or more for high income economies. The data was obtained from the World Health Organization (2012).

lower middle income countries, cardiovascular conditions catch up (26.5%), but are still almost matched in frequency by infectious diseases (23.3%). In upper middle (8.5%) and high income countries (6.7%), the importance of infectious disease while weakened remains accountable for a significant number of deaths. Globally, infectious diseases are the second most frequent cause of death (18.3%), even more prevalent than all forms of cancer combined (15.8%) and only preceded by cardiovascular diseases (33.7%).

Focusing only on deaths caused by infectious disease, lower respiratory infections are most frequent (for each income region individually, low to high: 28.7%, 30.8%, 43.5% and 57.7% as well as worldwide: 34.5%; cf. figure 1.2). Diarrhoeal diseases and HIV/acquired immune deficiency syndrome (AIDS) are the next most common worldwide (16.9% and 17.3%, respectively) where diarrhea is more prevalent in lower income regions (16.6% and 21.4% versus 7% and 5.6%), while HIV/AIDS plays a major role irrespective of income region (low to high: 20.4%, 13.3%, 26.2% and 11.3%).

Dealing with highly virulent pathogens and preventing their spreading requires a multi-pronged approach. First and foremost, etiology and routes of transmission have to be understood. Knowledge of vectors and natural reservoirs is of great importance as a first line of defense. In the case of plague, for example, insecticides killing fleas were successfully used as a prophylactic measure, as was controlling rat populations (Barnes 1990). Sanitary precau-

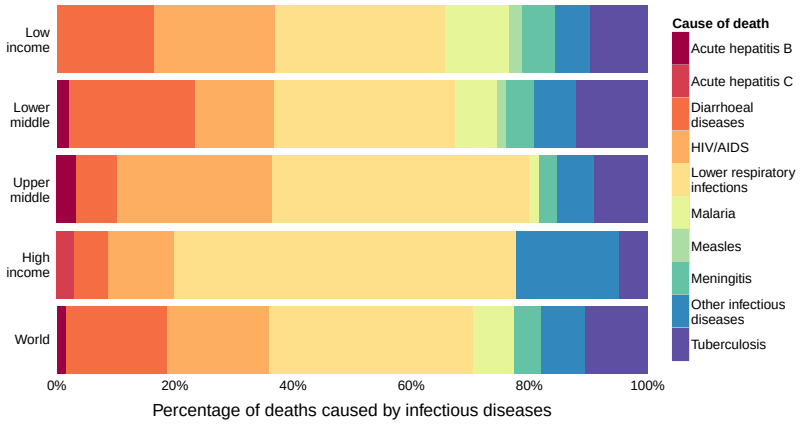


Figure 1.2: Relative frequencies of deadly infectious diseases for 2012 by World Bank income groups. Binning is based on gross national income (GNI) (see figure 1.1 for category thresholds). The data was obtained from the World Health Organization (2012).

tions including purification of drinking water, cocking foods well and the usage of disinfectants prevent initial infection, while measures such as sewage treatment, hand washing and wearing face masks help limiting spread among humans. Vaccination is the most important preventive measure. Exposing the immune system to a foreign antigen in a controlled manner artificially induces immunity. Among the great successes of widespread vaccination efforts is the global eradication of smallpox through a coordinated initiative lead by the World Health Organization in the 1970's.

Post-infection therapies include symptomatic treatments, as well as anti-infective drugs. Antibiotics exploit differences in proteomes between host and pathogen to selectively disable the invader with minimal toxicity to the host. This approach has been tremendously successful throughout most of the 20th century, leading to widespread application and prompting development of resistance towards the commonly used compounds. Adding to the severity of the problem is a lack of discovery of new drugs. No new class of anti-bacterial agents has been found since 1987, causing big pharmaceutical companies to withdraw from the area. The remaining research is mainly focused on improving on existing drugs, leading to a weak product pipeline, especially for the treatment of gram-negative bacteria (Silver 2011). In its first global study on anti-microbial resistance, the World Health Organization (2014) notes:

> Antimicrobial resistance (AMR) within a wide range of infectious agents is a growing public health threat of broad concern . . . A post-

> antibiotic era—-in which common infections and minor injuries can kill—-far from being an apocalyptic fantasy, is instead a very real possibility for the 21st century.

An alternative to pathogen directed search lies in targeting the set of host proteins necessary for infection. Many intracellular parasites subvert cellular functions to gain entry via host-mediated processes such as endocytosis. Upon entry, they move to a suitable niche and rely on host resources for proliferation. Challenges include evading host-cell defense mechanisms, generating sufficient space for replication, nutrient acquisition and keeping the host alive as long as possible, most of which require complex interactions between invader and host-based mechanisms. Finally, exiting the host cell again requires the parasite to successfully insert itself into existing signaling pathways (Leirião et al. 2004).

Host directed therapeutics (HDT) offer an escape from the conundrum of wanting to combat but not wanting to select for the surviving microbial parasites. The major challenge under this regimen is finding infectome components that are nonessential for cell survival, as orthogonality of host and infectant can no longer be exploited. Proving feasibility of the approach, Czyż et al. screened a library of 640 compounds already approved by the United Stated Food and Drug Administration (USFDA or short FDA) for inducing resistance to four intracellular pathogens (*Coxiella burnetii*, *Legionella pneumophila*, *Brucella abortus*, and *Rickettsia conorii*). They found multiple drugs, not classified as antibiotics, that successfully inhibited intracellular bacterial growth while entailing only limited toxicity to THP-1 host cells. Prussia et al. review the usage of genome-wide screens to study host-pathogen interactions (for HIV and influenza) which in turn serve as basis for rational identification of drug targets for novel host-directed antivirals.

A detailed understanding of the human infectome is of crucial importance to the development of HDT and may even benefit the development of new anti-microbial agents. Feasibility of systematic loss of function screens using ribonucleic acid (RNA) interference methodology offers a unique opportunity to investigate complex cellular networks, making this an ideal tool for laying groundwork in combating infectious diseases. Of great importance, however, is ensuring reproducibility and comparability of such datasets, as well as ready availability to the scientific community.

Created to tackle such issues, InfectX and TargetInfectX are two successive Research, Technology and Development (RTD) projects funded by SystemsX, the Swiss initiative for systems biology, contracted with identification and study of the human infectome for a set of bacterial (*Bartonella henselae*, *Brucella abortus*, *Listeria monocytogenes*, *Salmonella* typhimurium and *Shigella flexneri*) and viral

pathogens (adenovirus, rhinovirus and *Vaccinia virus*). The central effort of generating kinome- and genome-wide siRNA screens for each of the investigated pathogens and capturing image data followed by computational image analysis is carried out by an interdisciplinary consortium of research groups spanning the Universities of Basel and Zürich, the Swiss Federal institute of Technology (ETH Zürich), as well as the Pasteur Institute.

A wealth of data is easily generated by running computational feature recognition at single cell resolution on microscopic images which subsequently calls for sophisticated statistical analysis in order to expose biologically relevant information. This master thesis explores the single cell feature space, provides software to handle such datasets and attempts to employ generalized linear models (GLMs) for comparing siRNA-knockdown experiments and identifying discriminatory features. Following this introductory part, chapter 2 will provide the necessary biological background, covering microbial infection mechanisms, a description of each pathogen in terms of physical characteristics, diseases caused, pathogenesis and epidemiology, as well as giving an introduction on RNA interference, including molecular mechanism, biological function and several applications alongside methodological caveats. Chapter 3 reviews InfectX data collection, detailing experimental setup, image acquisition, data processing and image analysis, as well as qualitatively describing the available datasets.

An R package called singleCellFeatures was developed to facilitate working with single cell feature datasets, the specifics of which are detailed in chapter 4. Beginning with a motivational example, the importance of efficient computation and data representation is highlighted and the R package is described in terms of developed S3 classes as well as data fetching, caching, manipulation and analysis routines. Finally, chapter 5 introduces GLMs, parallel mixed models (PMMs), deliberates several normalization approaches and concludes with some analysis results along with a discussion of unresolved issues.

Chapter 2

Biological Background

In order to better understand infectious diseases from a cell biological standpoint, this chapter reviews the current state of knowledge surrounding both bacterial and viral entry mechanisms. A sweeping overview of epidemiology and pathogenesis for several specific bacterial (*Bartonella henselae*, *Brucella abortus*, *Listeria monocytogenes*, *Salmonella enterica* and *Shigella flexneri*), as well as viral parasites (adenoviruses, rhinoviruses and *Vaccinia virus*) is given and the chapter concludes with a look at RNA interference as this mechanism is a cornerstone of genome-wide knockdown experiments.

2.1 Microbial Host-Cell Infection

Multi-layered keratinized skin is impenetrable for almost all microbial parasites. Instead they either require breaches such as cuts, scratches, puncture wounds and arthropod bites, or environmental interfaces which offer less impervious protection. Examples include respiratory, gastrointestinal and urogenital tracts, which all contain segments where only a single layer of epithelial cells has to be overcome. Although often protected by chemical defense mechanisms (acidity of the stomach and urogenital tract, as well as microbicidal factors in mucous secretions in the respiratory tract and small intestine), combined with frequent flushing (urination, peristalsis and the coordinated beating of cilia), some microbes have adapted to survive these hostile environments.

For extracellular pathogens to successfully colonize epithelial linings, they must avoid being removed by cleansing mechanisms of the host. Many bacteria accomplish this by expressing adhesins, protein complexes that recognize and bind to specific host-cell receptors, providing host and tissue tropism. Bacterial

pili serve to extend reach and penetrate mucous secretions and therefore often carry adhesins. Enteropathogenic *E. coli* (EPEC) have extended this scheme by injecting their own receptor protein Tir (translocated intimin receptor) through the T3SS into the host cell to which it then attaches. This has entails the additional convenience that the intracellular domain of Tir can be used to modify host cell behavior (Alberts et al. 2008).

The outside of many epithelial barriers is covered in natural bacterial flora and crossing over into sterile cavities has the advantage of not having to compete with organisms well accustomed to that particular niche. Furthermore, intracellular pathogens are no longer accessible to antibodies and phagocytic cells and perhaps have a nutrient rich environment at their disposal.

2.1.1 Viral Infection Mechanisms[1]

The first step of any viral entering sequence is binding to the target cell surface. This can be mediated by attachment factors which simply serve to concentrate the virions on the cell surface or by virus receptors, which additionally act as communicators between host and pathogen. Common attachment factors include glycosaminoglycan chains and sialic acids and are comparatively unspecific. Glycoprotein spikes on enveloped and capsid proteins of non-enveloped viruses provide host specificity by binding cellular receptors. These cellular receptors typically serve other purposes and are exploited for infection. Binding affinity for individual interactions may be weak but aggregation of multiple interactions provide virtually irreversible avidity (Smith 2012).

Viral import. For viral cell entry, different strategies exist. Enveloped viruses can either directly fuse with the plasma membrane (e.g. HIV) or be endocytosed by the host cell (e.g. influenza), while non-enveloped viruses either create a pore and directly inject their genome into the cytosol (e.g. polio virus) or are endocytosed (e.g. adenovirus). Endocytosis has major advantages over alternative strategies. Reaching its replicatory niche within the host cell is a difficult task for a microorganism having no means of locomotion, and hijacking the endocytic system solves this problem elegantly. Furthermore, maturation of endosomes provides precise environmental cues to the invader for triggering uncoating and release. Both fusion with the cell membrane and injection of viral material into the cytosol leaves back traces of infection to be detected by the immune system. Being completely engulfed by the host, however, the intruders leave back no telltale traces. Additionally, lytic membrane penetra-

[1] Much of the information presented in this chapter is compiled from the online virus encyclopedia ViralZone, provided by the Swiss Institute of Bioinformatics (Hulo et al. 2011).

tion techniques are not as problematic to the host if only applied inside an endosome as opposed to the plasma membrane.

Endocytic viruses trigger uptake either in a receptor mediated fashion (clathrin, caveolin or lipid raft dependent) or via non-specific macropinocytosis. The clathrin pathway is most widely used, for example by rhinoviruses, some adenoviruses, and coronaviruses and presents with characteristic invaginations, termed clathrin-coated pits (CCPs). The inwards facing pockets are subsequently pinched off by the two membrane scission proteins dynamin-1 and dynamin-2, releasing clathrin-coated vesicless into the host cytosol. Caveolin-mediated endocytosis is thought to be a more tightly regulated and low capacity pathway but is nevertheless exploited by several virus species, including picornaviruses and some retroviruses. Caveola formation is lipid raft dependent and recruitment of caveolin yields 50–70 nm flask-shaped pockets which are closed off by dynamin action. A lipid raft dependent, caveolin independent pathway has been described in simian virus 40 infection, but remains poorly understood. Lastly, larger virions such as poxviruses or herpesviruses initiate macropinocytosis, a mechanism typically employed by the cell for non-specific uptake of extracellular particles. An actin dependent membrane ruffling leads to formation of a lamellipodium which folds back onto the plasma membrane, enclosing an extraluminal volume and thus creating a macropinosome.

Upon endocytic uptake, viral pathogens need to uncoat and eject their genetic material into the cytosol, as soon as their replicatory niche is reached. Escape timing is a critical issue, as late endosomes turn into lysosomes, capable of digesting their contents. Many enveloped viruses employ fusion mechanisms, which can be classified as type I or type II. For both types, increasing acidity associated with endosome maturation, initiates membrane fusion. Type I fusion proteins are forced into a metastable conformation prior to being added to the viral envelope and low pH triggers a conformational change to a state of lower energy. The energy released is used to force the two membranes close together resulting in their fusion (see figure 2.1). In type II fusion proteins, the critical transformation is not a conformational change but one in quaternary structure (Harrison 2008).

Non enveloped viruses cannot fuse with host membranes and have developed alternative approaches such as lysis (e.g. adenovirus) or ejecting their genome through pore-forming complexes (e.g. reovirus). Polyomaviruses need to pass through the endoplasmic reticulum (ER) because they rely on ER localized proteins to uncoat their capsid. For export from the ER into the cytosol, they exploit the ER-associated protein degradation pathway, which serves as export mechanism for misfolded proteins from the endoplasmic reticulum to be degraded by proteasomes (Smith 2012).

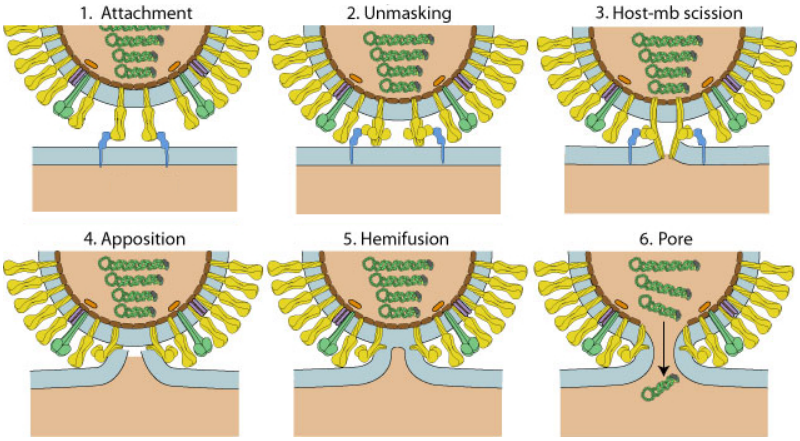


Figure 2.1: A generalized view of the steps necessary for viral–cellular membrane fusion. In pre-fusion conformation, fusion proteins have their hydrophobic fusion moieties tucked away. Upon attachment (1), they are unmasked (2), interact with the target membrane and ultimately penetrate it. Conformational change in fusion proteins induces membrane scission (3) and forces the two bilayers into close proximity (4), yielding a state of hemifusion (5). Finally a fusion pore is formed (6), stabilized by the post-fusion conformation which is lower in energy than the pre-fusion state. Adapted from Hulo et al. (2011).

Table 2.1: The Baltimore classification scheme is based on diversity of genetic system that have evolved in viruses. For each group, a selection of virus families capable of infecting humans, is provided, along with whether the virions are enveloped and the location of their replicatory niche. The data is compiled from Hulo et al. (2011).

	Genome based class	Examples	Enveloped	Replication site
DNA viruses	Group I: dsDNA	*Adenoviridae*	no	nucleus
		Poxviridae	yes	cytoplasm
	Group II: ssDNA(+)	*Parvovirinae*	no	nucleus
		Anelloviridae	no	nucleus
RNA viruses	Group III: dsRNA	*Reoviridae*	no	cytoplasm
	Group IV: ssRNA(+)	*Coronaviridae*	yes	cytoplasm
		Picornaviridae	no	cytoplasm
		Hepeviridae	no	cytoplasm
	Group V: ssRNA(-)	*Filoviridae*	yes	cytoplasm
		Paramyxoviridae	yes	cytoplasm
Retro	Group VI: ssRNA(+)-RT	*Orthoretrovirinae*	yes	nucleus
	Group VII: dsDNA-RT	*Hepadnaviridae*	yes	nucleus

Replication. In contrast to larger intracellular parasites that carry the genetic information required for sustaining their own metabolism and replication, viral pathogens typically rely almost exclusively on host machinery. Furthermore, viruses have developed strategies for interfering with host transcription and translation in order to promote synthesis of viral proteins at the expense of host gene expression. Even modulation of the host cell cycle is not uncommon, as some deoxyribonucleic acid (DNA) viruses (including adenoviruses) are able to trigger a G1 to S phase transition, yielding an increased concentration of active DNA polymerase, while other species are capable of inducing a G2/M arrest, which again provides an optimized environment for those viruses. Further virus–host interactions include regulation of apoptosis, immune response modulation and interferon signaling.

The remarkable diversity of genomic systems employed by viruses is captured by a classification system devised by Baltimore (1971). Table 2.1 lists the 7 types of viral genomes alongside examples of human viruses for each group, as well as whether those viruses are enveloped and where they replicate. Consequently, requirements for replication, transcription and translation vary widely among the different groups of viruses and due to the resulting mechanistic heterogeneity, viral propagation is not further explored within this general section. An excellent overview is provided by the online database of Hulo et al. (2011).

Viral export. The final stage of the viral life-cycle is concerned with virion assembly and exiting the host cell. Again, many strategies exist. Some nuclear replicating viruses (such as polyomaviruses) assemble their capsid proteins within the nucleus, requiring their structural proteins to target the nucleus via nuclear localization signals and leave the nucleus by disrupting the nuclear envelope, while others have their genome exported via nuclear pores and assemble progeny virions in the cytoplasm. Some cytoplasmic viruses (including poxviruses) replicate within special structures called viral factories or viroplasms, which increase efficiency of assembly and packaging and provides protection from host defense mechanisms. Other cytoplasmic viruses localize to organelles such as the ER (e.g. flaviviruses) where they are assembled and enter the secretory pathway via the Golgi apparatus. For intracellular motility, large virions such as poxviruses or herpesviruses have to rely on microtubule dependent transport whereas particles smaller than 20 nm can freely diffuse within the cytosol.

Once the host cell resources are depleted and replication is completed, progeny virions trigger their release. Viral shedding may occur via cell lysis, apoptosis, exocytosis or virion budding. Most non-enveloped and few enveloped viruses disrupt the plasma membrane with lytic viroproteins leading to cell death and release of cytoplasmic contents. While many viruses inhibit apoptosis, typi-

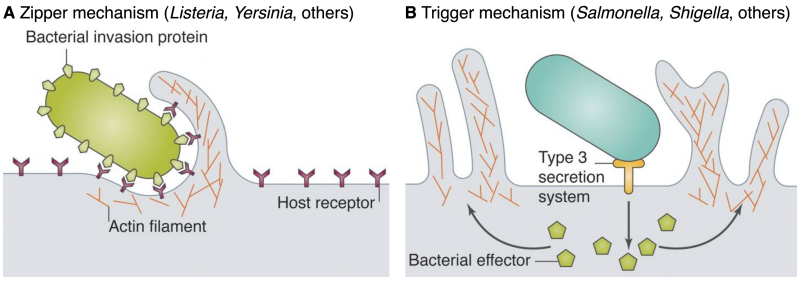


Figure 2.2: Both the zipper (A) and trigger mechanisms (B) are actin dependent and lead to phagocytosis by usually non-phagocytic host cells. Zippering bacteria display an invasion protein on their surface that recruits actin filaments via a host receptor, while triggering bacteria inject an effector into the host cytosol by means of a syringe like type III secretion system leading to their uptake. Adapted from Haglund and Welch (2011).

cally employed as a host defense measure, some (including hepeviruses and lentiviruses) have been implicated in exploiting this mechanism for expulsion and possibly subsequent infection of macrophages. Exocytosis and virion budding are two release strategies that are non-lethal to the host cell. Enveloped viruses acquire host-derived membrane either within the cell, typically at ER or Golgi exit sites or directly from the plasma membrane. In the latter case, envelopment coincides with host exit, whereas in the former case, virions are expelled via fusion of exocytic vesicles with the plasma membrane.

2.1.2 Bacterial Entry Mechanisms

Due to the much larger size of bacterial pathogens, endocytosis is not a feasible mechanism for entry, whereas phagocytosis can deal with uptake of particles this large. While phagocytosis is a function usually only available to macrophages, some bacteria have evolved mechanisms of inducing phagocytosis in other cell types. Species explicitly targeting macrophages, such as *Mycobacterium tuberculosis* and *Legionella pneumophila*, have to be able to escape phagosomes or deal with resisting digestion.

Two recurring patterns for inducing phagocytosis in non-phagocytic cells have been described by Cossart and Sansonetti (2004) as zipper (encountered in *Yersinia pseudotuberculosis* and *Listeria monocytogenes*) and trigger mechanisms (used by *Salmonella enterica* and *Shigella flexneri*). Not all entry strategies can be assigned to these two classes and several additional, unrelated pathways have been uncovered.

Zipper mechanism. Host entry by zippering bacteria is characterized by bacterial surface proteins binding to cellular receptors, thereby inducing signaling cascades that lead to limited, localized actin rearrangements. This extends the plasma membrane alongside the entering bacteria in a zipper-like fashion. Cossart and Sansonetti divide the scheme into three successive steps:

(a) Contact and adherence. Independent of the actin cytoskeleton, bacterial adhesins interact with host cell surface proteins leading to receptor clustering. *Y. pseudotuberculosis* displays invasin, capable of interacting with β_1 integrins, while *L. monocytogenes* uses InlA, a protein that binds E-cadherin. Cadherin and integrins are usually involved in anchoring cell junctions and the invading bacteria mimic initiation of junction formation by a neighboring cell.

(b) Phagocytic cup formation. Responding to the mistaken signal, the target cell extends its surface towards the signal origin via Arp2/3 (actin-related-protein 2 and 3) and Rac1 (Ras-related C3 botulinum toxin substrate 1) mediated actin polymerization, attempting to cover the adhesive surface. This leads to engulfment of the invading bacterium.

(c) Phagocytic cup closure. In a process probably involving the phosphotransferase phosphatidylinositol-4-phosphate 5-kinase, which catalyzes the conversion of PI(4)P (phosphatidylinositol 4-phosphate) to PI(4,5)P_2 (phosphatidylinositol 4,5-bisphosphate), the protruding membrane sections fuse and actin depolymerization returns the plasma membrane to its original state.

Trigger mechanism. In trigger schemes, bacterial T3SS weakly adhere to target cell receptors and inject effector molecules into the host cytosol through a pore formed by the syringe like delivery mechanism. These proteins directly interact with actin regulatory cellular components, causing major actin rearrangement characterized by membrane ruffling. This in turn leads to non-specific engulfment of bacteria and surrounding particles. Four steps have been identified by Cossart and Sansonetti:

(a) Pre-interaction stage. Effector molecules, stored in bacterial cytosol, are associated with dedicated chaperones in order to prevent aggregation, degradation and once secretion is initiated guide them towards T3SS. T3SS needle complexes (several hundred per bacterium are possible) are fully assembled and integrated into the two membranes of gram-negative bacteria.

(b) Interaction stage. The tip of T3SS recognizes the host cell membrane and activates secretion via a process not well understood. Originally it was be-

Table 2.2: A selection of T3SS genes in *Salmonella* and *Shigella*. The *Salmonella* genome encodes two separate secretion systems which are deployed at distinct phases of infection. The data was obtained from Wang et al. (2012).

T3SS component	*Salmonella* SPI-1	*Salmonella* SPI-2	*Shigella*
Outer membrane ring	InvG	SsaC	MxiD
Inner membrane ring	PrgK, PrgH	SsaJ, SsaD	MxiJ, MxiG
Cytoplasmic ring	SpaO	SsaQ	Spa33
Export apparatus	SpaP, SpaQ, InvA	SsaR, SsaS, SsaT	Spa24, Spa9, MxiA
Needle assembly	InvI, InvJ, OrgA	SsaO, SsaP, SsaK	Spa13, Spa32, MxiK
Needle major subunit	PrgI	SsaG	MxiH
Needle minor subunit	PrgJ	SsaI	MxiI
Translocon	SipB, SipC, SipD	SseB, SseC, SseD	IpaB, IpaC, IpaD
ATPase	InvC	SsaN	Spa47
Effector export	InvE		MxiC
Chaperone	SicA, SicP, InvB	SseA, SscA, SscB	IpgC, IpgE, IpgA
Secreted effector	AvrA, SipA, SptP, SopB, SopE, SopE2	SifA, SifB, SpvC, SseG, SspH1, SspH2	IpaA, IcsB, IpgD, IpgB1, OspG, VirA
Transcription regulator	InvF, HilA, HilC, HilD, HilE, PhoP	SsrA, SsrB, OmpR, H-NS, Hha	MxiE, VirB

lieved that the needle complex was able to puncture the target membrane but recent evidence suggests that translocator proteins are ejected which subsequently form a pore and thusly facilitate insertion of the secretion system. Once bacterial and cellular cytoplasms are connected, effector chaperones dissociate and unfolded proteins (the needle passage is only 3 nm wide) enter the host in a probably adenosine triphosphate (ATP) dependent manner. Chaperones serve a double purpose as transcription factors, encouraging synthesis of new effector protein when not attached to their substrate.

(c) Formation of macropinocytic pocket. Massive, but localized membrane protrusions emerge due to action of bacterial effectors. In *Shigella* entry, virulence spread protein A (VirA) causes local destabilization of microtubules by binding to α/β-tubulin heterodimers. This in turn stimulates Rac1 activity, Cdc42 (cell division control protein 42 homolog) recruitment and subsequent Arp2/3 activation leading to actin polymerization. *Shigella* invasion protein C (IpaC) recruits the Src tyrosine kinase further

enhancing actin dynamics. *Salmonella* inject the proteins *Salmonella* outer protein E (SopE) and SptP, an activator/inhibitor pair for the GTPase complex Rac1/Cdc42. Guanine nucleotide exchange factor (GEF) activity of SopE induces actin rearrangements leading to formation of the macropinocytic cup.

(d) Closing of macropinocytic pocket. In *Salmonella* invasion, GTPase activating protein (GAP) activity of *Salmonella* protein tyrosine phosphatase (SptP), restores the inactive guanosine diphosphate state of Rac1/Cdc42 and leads to actin depolymerization. Its GEF partner SopE is degraded more rapidly than SptP, enabling reversible control over the pathways exploited for entering. In case of *Shigella*, actin depolymerization is initiated by binding of IpaA to vinculin, a key protein of focal adhesion plaques.

Other entry pathways. In addition to trigger and zipper type uptake, other atypical mechanisms exist. Host cell entry of *Brucella abortus*, for example, has been described as invasome mediated (Dehio 2005). In this actin dependent process, bacteria aggregate on the cell surface and trigger their engulfment by injecting bacterial effectors into the host via T4SS. The internalized structure is called an invasome.

Actin-independent uptake albeit rare, is possible as evidenced by *Campylobacter jejuni* and *Citrobacter freundii*, which have evolved a microtubule dependent invasion strategy (Kopecko, Hu, and Zaal 2001). A further example of microtubule involvement is presented by *Clostridium* spp. (a genus that includes the etiological agent of tetanus and several species capable of botulinum toxin synthesis). These intercellular pathogens induce formation of long protrusions formed by microtubule filaments that wrap around the bacteria and fix them in close proximity. It has been speculated that such a mechanism could be exploited by intracellular pathogens to promote adherence (Haglund and Welch 2011).

2.1.3 Intracellular Survival[2]

While bacteria that successfully subvert endocytic pathways and trigger their uptake have defied innate and evaded adaptive immune responses, they are still faced with defensive mechanisms by their new hosts. A multitude of strategies has evolved in order to undermine hostile actions directed at intruding bacteria and for establishing a replicatory niche within this initially adverse but nevertheless potentially propitious environment.

[2]Each subsection within this section is based on one or two review articles, referenced at the end of the first paragraph, respectively.

Phagocytic vacuoles, containing internalized microorganisms are destined for endocytic maturation. These compartments undergo successive acidification and finally develop into mature degradative phagolysosomes. Most intracellular pathogens either escape the endocytic vacuole before fusion with lysosomes occurs or manipulate cellular pathways that control its maturation, thereby creating a niche permissive to their survival. Two important themes for cytosolic bacteria are efficient means of locomotion and evasion of cellular responses such as autophagy (Alberts et al. 2008).

The phagocytic vacuole. In order to survive an environment characterized by constantly decreasing pH, poor nutrient content and an increasing concentration of antibacterial and lysosomal enzymes, most successful pathogens incapable of escaping their internalization compartment alter biogenesis and dynamics of their surroundings. Examples include *Salmonella*, *Mycobacterium tuberculosis* and *Legionella pneumophila* (Ham, Sreelatha, and Orth 2011; Ray et al. 2009).

Salmonella replicates in plasma membrane derived perinuclear *Salmonella*-containing vacuoles (SCVs) that are neither early, nor late endosomes. While some fusion events with vesicles of the endocytic pathway take place, maturation into lysosomes is inhibited. SopB, an SPI-1 encoded T3SS effector, hydrolyzes $PI(4,5)P_2$ and reduced concentration of $PI(4,5)P_2$ inhibits recruitment of cellular RAB GTPases required for phagosome-lysosome fusion. A key role in manipulating membrane trafficking is played by the SPI-2 encoded T3SS effector *Salmonella*-induced filament A (SifA). Endosomal maturation is associated with microtubule dependent vacuole relocation and SifA has been shown to regulate kinesin activity, thereby contributing to SCV integrity. Furthermore, SifA activity is essential in increasing the SCV volume and creating the necessary space for replication.

Similarly to SopB in *Salmonella*, the bacterial phosphatase SapM expressed by *M. tuberculosis*, dephosphorylates phosphatidylinositol 3-phosphate which blocks phagosomes from fusing with late endosomes and consequently arrests maturation. A different approach is employed by *L. pneumophila* which secrete SidC via T4SS, an effector protein capable of binding and displaying PI(4)P on the vacuolar surface. This leads to recruitment of ER-derived vesicles which turns the phagosome into an LCV (*Legionella*-containing vacuole) which is removed from the endocytic pathway and becomes sufficiently spacious for replication.

Cytosolic replication. Pathogens that evolved to replicate in the cytosol must quickly escape the internalization vacuole. Most species are capable of triggering their release within 30 minutes of infection, highlighting the importance

of quick action in order to prevent damage incurred by acidification of the phagosome. In all known cytosolic bacteria vacuolar egress is a pathogen driven process and most species employ lytic enzymes capable of forming trans-membrane pores. Examples of bacteria that enter the cytosol as part of their life-cycle include *Listeria monocytogenes*, *Shigella flexneri*, *Burkholderia pseudomallei*, *Francisella tularensis* and species of the *Rickettsia* genus (Ray et al. 2009).

The best studied organism, *L. monocytogenes*, secretes listeriolysin O (LLO) a hemolytic enzyme, capable of inserting into the target membrane, oligomerize and through a conformational change form a pore. This delays endosomal maturation, prevents fusion with lysosomes and finally releases the bacteria into the cytosol. Regulation of LLO is pH mediated and involves the host factor γ-interferon-inducible lysosomal thiol reductase, specific to endosomes, phagosomes and lysosomes. Vacuole escape by *S. flexneri* is dependent on the three T3SS translocator proteins IpaB, IpaC and IpaD with IpaB and IpaC forming a pore complex and IpaD facilitating insertion into the membrane. For both *Rickettsia* spp. and *B. pseudomallei*, little information on vacuole-lytic mechanisms exist. Hemolysin C and phospholipases have been implicated of playing important roles but more work is necessary to uncover their exact mechanistic relevance. *F. tularensis* stands out among the previously mentioned organisms in that lysis of the internalization vacuole is followed by re-entry of another membrane bound compartment, the *Francisella*-containing vacuole.

Interestingly, it is not known to what extent, the nutritional content of mammalian cytosol is permissive to bacterial growth. As evidenced by *S. flexneri* which can grow with a doubling time of only 40 minutes (growth rates in laboratory medium are comparable) it is possible for microorganisms to adapt to this environment. While it seems natural to assume that cellular cytosol provides ideal conditions for bacterial growth and replication, this raises the question of why only comparably few pathogenic organisms exploit this ecological niche. Indeed, nutritional arguments are not the only reasons to consider, as it is clear that reaching this habitat is no easy feat and further defense against foreign particles in the cytosol, for example autophagocytosis, is employed.

Actin based motility (ABM). As for large viruses, free diffusion within the cytoplasm is not readily possible for bacteria and a feature common to most cytosolic bacterial is actin based motility (ABM). Actin monomers exist in two forms, G-actin (globular) and F-actin (filamentous). ATP binds G-actin, and upon formation of a trimeric nucleus the nascent chain grows in both directions while ATP is hydrolyzed. When the chain reaches a certain length, growth becomes directional and association of monomers at the plus end compensates for dissociation of monomers at the minus end. This leads to a self-sustaining

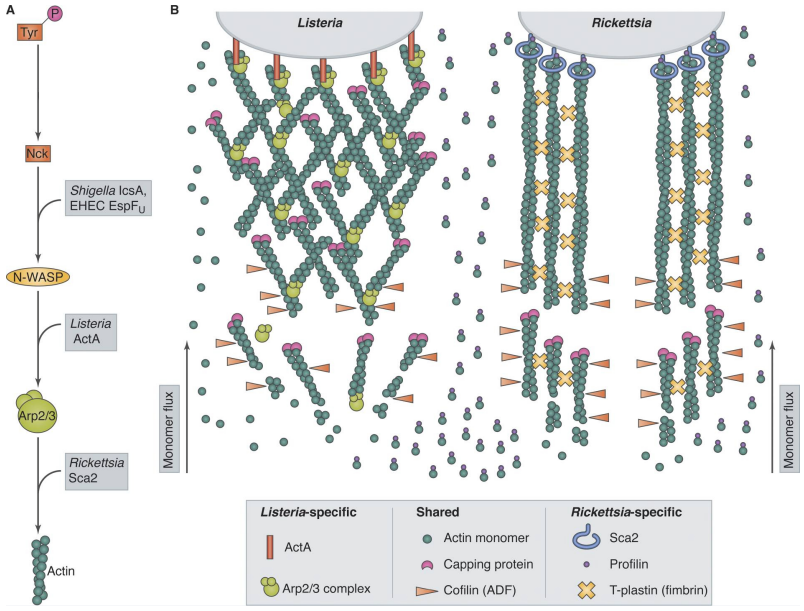


Figure 2.3: The capability for actin based motility (ABM) evolved independently in several intracellular bacterial pathogens. (A) The entry points into the cellular actin assembly pathway vary, with some organisms providing their own nucleation promoting factor (NPF) (*Listeria* and *Rickettsia*), while others rely on cellular NPF (*Shigella* and enterohemorrhagic *E. coli*). The branching pattern of actin tails differs between organisms relying on cellular Arp2/3 (actin-related-protein 2 and 3) from those capable of Arp2/3 independent ABM (B). Cofilin is responsible for actin depolymerization, profilin is a regulatory protein catalyzing the exchange of ADP to ATP and fimbrin crosslinks actin strands into bundles. Adapted from Haglund and Welch (2011).

scheme described as treadmilling. Limiting to actin polymerization is the kinetically unfavorable nucleation step and in vivo, this is stimulated by cellular factors such as Arp2/3 which require activation by nucleation promoting factors (NPFs), including members of the Wiskott-Aldrich syndrome protein (WASP) family (Stevens, Galyov, and Stevens 2006; Haglund and Welch 2011).

ABM exhibiting pathogens can be divided into two groups depending on whether they provide NPFs of their own or recruit host cell NPFs. Expression of ActA (*Listeria* actin assembly-inducing protein) by *L. monocytogenes* is both necessary and sufficient for ABM as shown in a cell free system using ActA coated beads, actin, Arp2/3, CapZ, cofilin and ATP. No cellular NPF is required as ActA mimics WASP and directly activates Arp2/3. *Rickettsia* are among the few pathogens capable of ABM without relying on Arp2/3, leading

to formation of actin tails with a distinct morphology. The formin-like protein Sca2 directly interacts with actin and RickA, exhibiting some homology to WASP can act as NPF. A further organism that mimics cellular NPF is *B. pseudomallei*, which is capable of BimA-mediated, Arp2/3 independent actin polymerization.

Bacterial pathogens that rely on cellular NPF instead of supplying their own include *S. flexneri*, *Mycobacterium marinum* and enterohemorrhagic *E. coli* (EHEC). In *S. flexneri* infection, bacterial *Shigella* intracellular spread protein A (IcsA), capable of recruiting neural Wiskott-Aldrich syndrome protein (N-WASP), is the only required bacterial factor for triggering ABM. By inducing conformational changes in N-WASP, IcsA is thought to mimic activation by Cdc42, allowing for association of N-WASP with Arp2/3. Details of ABM in *M. marinum* are less well known but as it is both WASP and Arp2/3 dependent, it is assumed that the employed mechanism is similar to that of *S. flexneri*.

Both EHEC and EPEC, while not intracellular pathogens, are able to induce cytosolic actin polymerization. By inserting the transmembrane protein Tir into the plasma membrane they are able to initiate actin polymerization via the same pathway as ABM, leading to the formation of pedestals protruding outwards. This is thought to provide means of extracellular motility. Intercellular motility by ABM capable bacteria is achieved by pushing against the host cell membrane into the neighboring cell body until being engulfed in a double membrane vacuole that is subsequently lysed.

Autophagy. The basic catabolic cellular mechanism of autophagy is initiated as response to stress signals such as nutrient starvation, damaged cellular compartments and foreign particles. Cytoplasmic components marked for degradation are captured by the double-membrane autophagosome, intended for fusion with lysosomes. As cellular defense mechanism, cytosolic pathogens are targeted by selective autophagy in a process also related to as xenophagy. Bacterial response can be categorized as interference with the autophagy pathway, evasion of autophagy recognition or escape from the autophagosome (Ham, Sreelatha, and Orth 2011; Huang and Brumell 2014).

S. typhimurium, while not usually considered a cytosolic pathogen, has been shown to occasionally escape the phagocytic vacuole and replicate in the cytosol. Membrane damage triggers an amino acid starvation response that activates autophagy signaling by relocating mechanistic target of rapamycin (MTOR) from the late endosome to the cytosol. The bacteria respond via unknown action probably involving SPI-2 T3SS secretions that are able to restore both the cytosolic amino acid pool and MTOR localization, successfully inhibiting autophagy. Further examples of autophagy-initiation signaling inhibition

are provided by *M. tuberculosis* which is able to produce enhanced intracellular survival protein, an inhibitor to production of reactive oxygen species needed as an autophagy signal. Furthermore, bacterial toxins apt to interfere with cyclic adenosine monophosphate (cAMP) regulation can block autophagy in mammalian cells.

After autophagy has been initiated, the procedure can still be modified as shown by *L. pneumophila* and *S. flexneri*. The former organism secretes RavZ via T4SS which decouples the autophagy marker LC3 from the phagosomal membrane while the latter expresses VirA via T3SS, a GAPcapable of inactivating the autophagy regulator RAB1. Another group of bacteria prevent or delay lysosome fusion and accumulate in non-degradative vacuoles at neutral pH. Examples include *M. marinum*, *Chlamydia trachomatis*, *Yersinia pestis* and *Helicobacter pylori*.

Evasive measures have been suggested to be employed for *S. flexneri* and *L. monocytogenes*. IcsB of *S. flexneri* and ActA of *L. monocytogenes* have been implicated of masking the bacteria from autophagy targeting mechanisms and it has been suggested that ABM might somehow facilitate hiding from host cell detection. By also currently unknown means, *B. pseudomallei* can escape the phagosome of an autophagy-related process, possibly involving BopA secretion via T3SS.

2.2 Select Bacterial Pathogens

A total of 5 bacterial pathogens were selected for study within the InfectX RTD project by SystemsX. This section shortly describes each organism in terms of microbiological features, pathogenesis, epidemiology and diseases caused in humans. For each organism, a chapter of Gillespie and Hawkey (2006) serves as basis and is supplemented by one or two review articles referenced in the first paragraph of each section.

2.2.1 *Bartonella henselae*

Bartonella henselae is a short, rod shaped, unflagellated proteobacterium, phylogenetically closely related to the genus *Brucella*, presenting 94.4% 16S ribosomal RNA (rRNA) gene sequence homology, compared with *Brucella abortus*. The Gram-negative bacillus is a facultative anaerobic, intracellular parasite and was first described in 1992. Relatively harmless for healthy humans, infections can become life threatening in immunocompromised patients, making the species an important opportunistic pathogen (Anderson and Neuman 1997; Harms and Dehio 2012).

Diseases. In immunocompetent humans, infection with *B. henselae* can lead to a condition known as cat scratch disease (CSD). As the name suggests, most patients report being in contact with a cat and transmission often occurs through scratches and bites. Affecting primarily children and young adults (80% are 21 or younger), the self limiting infection typically presents itself with lymphadenopathy. Most patients remain afebrile and do not report feeling ill, with low-grade fever and malaise shown in roughly 30% of the cases. Recovery from uncomplicated CSD usually takes 2 to 6 months and requires no specific treatment.

Possible complications include Parinaud's oculoglandular syndrome (granulomatous conjunctivitis in one eye and parotid lymphadenitis on the same side), splenomegaly and hepatic or splenic abscesses, accompanied by fever, weight loss, fatigue and malaise. In 1 to 7% of the cases, the disease spreads to the central nervous system, leading to encephalopathy, but recovery is usually rapid (within several weeks).

Infections with *B. henselae* tend to have more severe consequences for immunocompromised patients, such as bacillary angiomatosis, bacteremia and endocarditis. AIDS patients suffering from CSD usually experience severe, progressive disease with infection spreading systematically and without appropriate treatement, fatal outcome. *Bartonella* spp. are the only prokaryotes known to be able to induce angiogenic tumors such as bacillary angiomas, which may involve skin, respiratory or gastrointestinal epithelia, hart, liver, spleen, bone marrow, muscles, or lymph nodes. Bacteremia may lead to inflamed heart valves, usually requiring endocarditic patients to have heart valve replacement surgery.

Pathogenesis. *B. henselae* are capable of intracellular growth in both epithelial cells and erythrocytes but the focus of this section lies on infection of the former cell type. Initial attachment is mediated by the trimeric autotransporter adhesin BadA which is capable of both interacting with the extracellular matrix and β_1-integrin, followed by effector secretion via the bacterial T4SS VirB/D4. For host cell entry, two mutually exclusive mechanisms have been described. Either single bacteria or small groups are phagocytosed via a zipper-like mechanism or large clusters are internalized in a unique cellular structure termed an invasome. While invasome formation is a slow process, taking 16 to 24 hours, *Bartonella*-containing vacuoles resulting from endocytosis are visible within minutes. It has been suggested that inhibition of endocytosis by either a combination of effector proteins BepC and BepF or the exclusive action of BepG is crucial to invasome formation as it allows for aggregation of bacteria on the cell surface. Not the activity of effector proteins but the clustering of cellular receptors may trigger large-scale internalization (see figure 2.4).

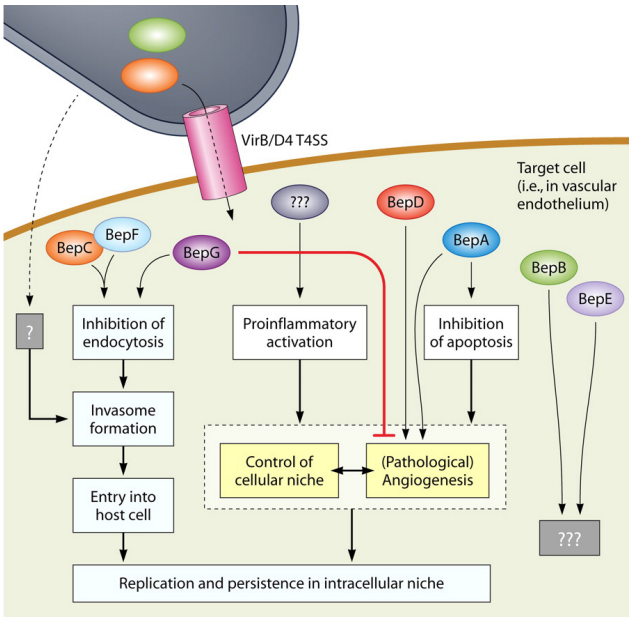


Figure 2.4: Several effectors secreted by a T4SS apparatus in *B. henselae* infection serve as virulence factors for colonization of the intracellular replicatory niche. The ones best characterized alongside their phenotype are schematically summarized. Adapted from Harms and Dehio (2012).

Pathological angiogenesis can be induced by a set of agonistic and antagonistic effectors. While BepG is a strong inhibitor to angiogenesis, both BepD (weakly) and BepA (strongly) promote sprouting of blood vessels. Although the exact mechanism of induction and regulation remains to be uncovered, the secretion of vascular endothelial growth factor (VEGF) has been demonstrated in vasoproliferative tumors caused by *B. henselae*. Furthermore, transcriptional promotion of various factors supporting angiogenesis, including intercellular adhesion molecule 1 (ICAM-1), interleukin-8 (IL-8) and angiopoietin-2 by bacterial effectors are unanimously agreed upon. Curiously, secretion of VEGF by infected endothelial cells has so far not been possible to show.

Inhibition of apoptosis is decisive to intracellular survival and it is assumed that BepA is capable of cAMP mediated antiapoptotic action. The role of two further effectors that have been identified as part of the T4SS in *B. henselae*, BepB and BepE, are unknown, as are the mechanisms leading to activation of proinflammatory signaling via NF-κB (nuclear factor κ-light-chain-enhancer of activated B cells).

Epidemiology. The role of cats and in particular kittens, as reservoirs to *B. henselae* has been firmly estabished. Infected felines are asymptomatic and show no signs of illness. Cat fleas (*Ctenocephalides felis*) serve as vectors to spread the bacteria among cats and have also been suspected of infecting humans. The main path of transmission to humans however, is through scratches and bites by infected cats. *B. henselae* has also been found in ticks and tick bites prior to contraction of CSD have been reported.

In the United States, 24000 cases of CSD are reported yearly, yielding 2000 hospital admissions with an estimated health care cost of $12 million. Children are more likely to be affected (80%) and incidence is higher in males (60%). The seasonal pattern (occurrences higher in fall/winter) is attributed to cat mating patterns, as well as pet acquisition fluctuations.

2.2.2 *Brucella abortus*

The Danish physician David Bang first isolated *Brucella abortus* in 1895 from cyetic cattle tissue, investigating a contagious disease causing abortions in cows. *B. abortus* are small, unflagellated proteobacteria with a cell wall consisting of an outer layer (9 nm) of lipopolysaccharide (LPS) and an inner layer (3–5 nm) of muramyl mucopeptide complexes. The Gram-negative cocobacilli appear to have evolved from free-living, soil-dwelling species and are closely related to other human pathogens such as *Bartonella* spp., based on 16S rRNA sequences. Brucella species were investigated for possible use as warfare agents in the mid 20th century by several armed forces. (Atluri et al. 2011; Bargen, Gorvel, and Salcedo 2012)

Diseases. Brucellosis is a human disease caused by several pathogenic *Brucella* species, most importantly *B. abortus*, *B. melitensis*, *B. canis* and *B. suis*. Onset may be acute or insidious and due to protean symptoms, diagnosis based on clinical presentation alone is difficult. The febrile disease is generalized and may involve many parts of the body, including nervous, skeletal, gastrointestinal, cardiovascular, respiratory and genitourinary systems. Furthermore, as the bacteria spread to other reservoir hosts via their reproductive systems, persistence of infection is crucial to the pathogen and it comes as no surprise that brucellosis can manifest as a chronic disease in humans too.

Fever is the most consistent sign of *Brucella* infection and depending on what specific organs are affected, further symptoms include asthenia, anorexia, nausea, malaise, arthritis, epididymo-orchitis in males, hepatomegaly, splenomegaly and pulmonary manifestations such as bronchitis or pneumonia. A rare complication (less than 2%), albeit the most lethal, is infective endocarditis. Invasion of the nervous system develops in less than 5% of cases and often results

in meningitis or meningoencephalitis with good prognosis under antimicrobial treatment.

Pathogenesis. Host entry occurs primarily via the digestive system but is also possible through the respiratory tract or skin lesions. Via the gastrointestinal route, *Brucella* spp. target Peyer's patches (lymphoid nodules localized towards the end of the small intestine) and must therefore pass through acidic conditions in the stomach. This is facilitated by expression of two ureases capable of hydrolyzing urea and producing a protective bicarbonate buffering system. When entering through the respiratory system, *B. abortus* target alveolar macrophages which serve as access point to the lymphatic system therefore facilitating systematic spread.

In order to persist at systemic sites, both active and passive mechanisms for evading the immune system are in place. LPS of the outer cell wall disguises the bacteria from toll-like receptors (TLRs) and expression of two proteins containing toll-interleukin-1 receptor domains actively interferes with TLR signaling.

Uptake by macrophages happens via phagocytosis, which is either triggered by nonopsonized bacteria through a lipid raft mediated mechanism or by opsonization. Although opsonin marked bacteria are 10-fold more likely to be ingested, the number of pathogens reaching their destination within the host cell is higher for nonopsonized bacteria. Still, most bacteria (up to 90%) are unsuccessful in evading their digestion and only very few are able to establish a replicative niche. Apart from professional phagocytes, epithelial cells may also be infected and the following paragraphs focus on this particular cell type.

Initial attachment is mediated by unknown eukaryotic receptors containing sialic acid residues that interact with *Brucella* surface protein. While involvement of bacterial HSP60 and and cellular prion protein has been proposed, this remains controversial. Maturation of early *Brucella*-containing vacuoles (BrCVs) is important for successful infection as preventing acidification (through addition of bafilomycin A) or fusion with lysosomes (through suppression of the late-endosomal GTPase Rab7) interferes with bacterial replication. This observation can be explained with acid serving as a trigger for expression of the T4SS VirB. However fusion events with late endosomes and lysosomes are only limited and under bacterial control.

Upon acquisition of late endosomal markers and acid initiated expression of T4SS, fusion with autophagic vacuoles occurs, leading to formation of an autophagosome-like compartment. Subsequent interactions with ER exit sites, mediated by secreted effectors, further modify the BrCV into an ER-derived vacuole, coated with ribosomal particles. At this stage, located within the ER-

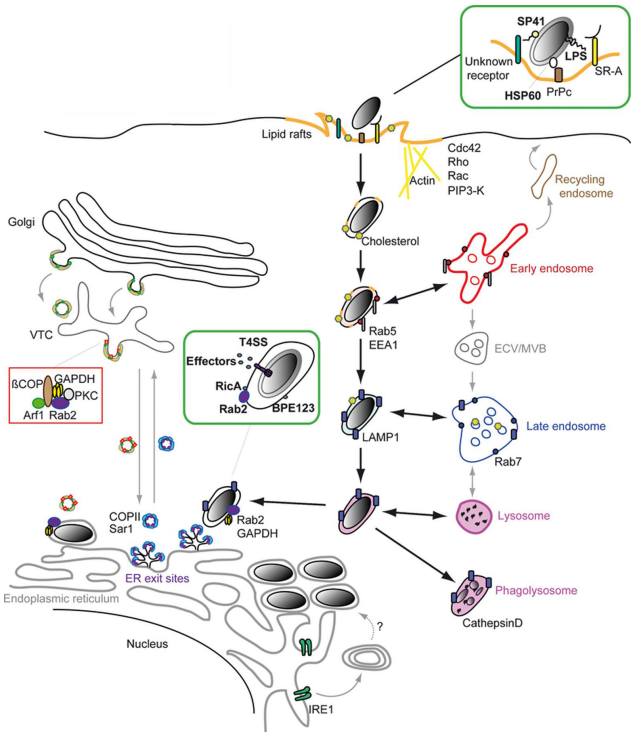


Figure 2.5: Schematic representation of the *B. abortus* intracellular life cycle, from cell entry via maturation of the *Brucella*-containing vacuole to establishing an intracellular replicatory niche. Interaction with many different cellular compartments of the endocytic pathway and the *cis* Golgi network are required for successful infection. Adapted from Bargen, Gorvel, and Salcedo (2012).

Golgi intermediate compartment, the replicatory niche is reached. Blocking the small GTPase Sar1 inhibits intracellular replication by preventing acquisition of coat protein complex II, participating in anterograde ER–Golgi transport, by ER-exit vesicles. Furthermore, the small GTPase Rab2, involved in ER–cis-Golgi traffic, is required for maximal proliferation, illustrating the dependence of *Brucella* on intercepting vesicular traffic.

Despite multiplying intra-cellularly in high numbers, host cells are kept alive and are even able to replicate despite infection. *Brucella* species are able to interfere with apoptosis, maintaining their replicatory niche, protected from immune response. It remains unknown what happens when the hosts capacity for freeloaders is reached, as well as how bacteria exit the host cell and spread.

Epidemiology. Preferred natural reservoir species for *B. abortus* are cattle (*Bos taurus* and *Bos indicus*) and almost all parts of the world are affected. The disease exists in both domestic and wild animals and is most prevalent in Mediterranean countries, North Africa, throughout the Middle East, India, Central Asia, as well as South and Central America. Zoonosis most often occurs through ingestion if unpasteurized milk products but airborne transmission is also possible, putting professionals involved in animal husbandry at risk. Vertical transmission among reservoir hosts can occur through lactation and horizontal transmission is facilitated by mating and placental discharge associated with aborted gestation. Human-to-human transmission is rare (but has been suspected to be possible via sexual intercourse), making humans dead-end hosts. As opposed to *Bartonella*, immunodeficient patients do not seem to be especially susceptible towards *Brucella* infections.

Worldwide, an estimated 500000 new cases of brucellosis occur annually, making it one of the most prevalent zoonoses. Although usually susceptible to combined antibiotic therapies of at least two agents (usually a tetracycline antibiotic combined with an aminoglycoside or rifampin), untreated brucellosis leads to a high degree of morbidity, leading to being classified a neglected zoonosis by the World Health Organization (WHO).

2.2.3 *Listeria monocytogenes*

The short, Gram-positive bacilli are non-sporeforming facultative anaerobes, capable of growing in a wide temperature range (0–50 °C) and in many different environments. Flagellation is temperature dependent with flagellin being expressed and assembled into peritrichous flagella around 20–25 °C but not at 37 °C. First described in 1924 by Murray after isolation from lymph glands of diseased laboratory animals, the pathogen was found to also infect humans four years later. For much of the time since, listeriosis was considered a rare zoonotic disease and did not receive much attention. It was not until the 1980s, when several food-borne listeria outbreaks caused a shift in interest towards the pathogen, which has since become a well studied facultatively intracellular parasite (Farber and Peterkin 1991; Cossart and Lebreton 2014).

Diseases. Maternal and neonatal listeriosis accounts for almost half of all infections. Listeriosis in pregnancy typically manifests in bacteraemia and presents as a self-limiting febrile disease with flu-like symptoms. Many cases, however are asymptomatic and the first sign of infection is abortion or neonatal listeriosis. Maternal infection does not necessarily carry over to the fetus, especially if proper chemotherapy is administered. Perinatal incidences are divided into early and late onset (>5 days after parturition), with former cases

typically resulting in septicemia and latter cases in meningitis. While in early onset cases the predominant route of transmission is transplacental, the situation is less clear in late onset cases. Both the maternal genital tract during child birth and environmental sources have been implicated. Despite antibiotic treatment, overall mortality rates of 30–40% are typical and prognosis for early onset disease is worse, as it is often associated with preterm birth and advanced stage of infection.

Among adults, most cases of listeriosis occur in T-cell deficient individuals. HIV infection, for example, increases incidence 150–300 fold compared to general population control groups. Predisposing conditions include lymphoreticular neoplasms, deliberate immunosuppression (e.g. antirejection treatment after organ transplants), alcoholism and diabetes mellitus. Despite increased susceptibility caused by immunodeficiency, roughly 30% of all infections affect immunocompetent individuals. In healthy adults, consumption of food contaminated with *L. Monocytogenes* can either lead to self-limiting febrile gastroenteritis with short incubation time (<24 h) or invasive listeriosis with much longer incubation periods (3–4 weeks). The systemic form of infection often manifests as bacteraemia or as a neurological infection, but can also involve endocarditis and spread to other parts of the body. Central nervous system involvement occurs in as much as 75% of cases and either presents as meningitis or encephalitis. Mortality rates of 35–45% have been reported for listeriosis in adults.

Pathogenesis. The predominant entry path for *L. Monocytogenes* into the human body is via the gastrointestinal tract, where Peyer's patches are targeted. The bacteria can induce cellular uptake, by non-phagocytic host cells via expression of cell-surface associated interanlins through a zipper-type entry program (see section 2.1.2). The cellular receptor for bacterial internalin InlA is E-cadherin and internalin InlB interacts with the receptor tyrosine kinase c-Met. Upon internalization, the phagosomal membrane is lysed, mediated by secretion of listerial haemolysin LLO in a cholesterol dependent mechanism whereby LLO monomers associate, oligomerize and form 35 nm pores. LLO is acid activated with an optimum around pH 5.5, which is reached in late endosomes. Moreover, LLO is required for autophagy modulation (see section 2.1.3) and has been implicated in regulating inflammatory response.

Growth and replication occur in the cytoplasm and ActA mediated ABM (see section 2.1.3) provides means of intracellular and intercellular movement. Adjacent cells can be entered by pushing against the plasma membrane and forming a pseudopod-like structure which in turn is taken up the neighbor. The resulting double-membrane vacuole is escaped by cytolysis, again dependent on LLO.

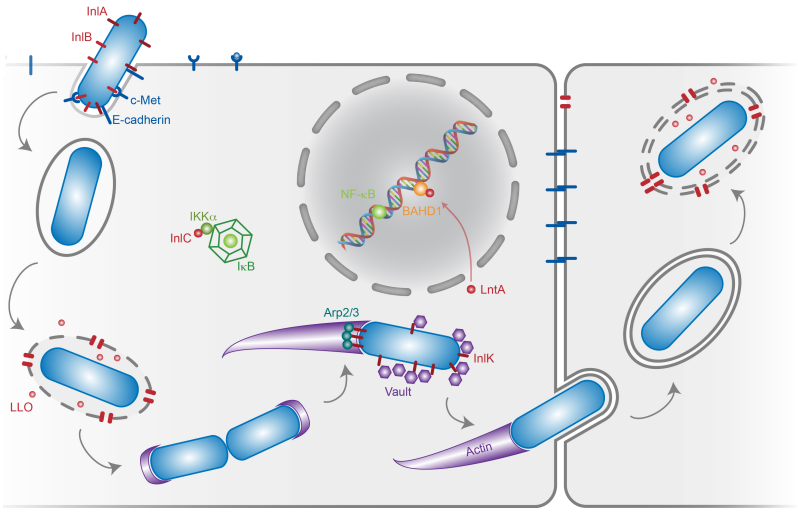


Figure 2.6: *Listeria* enter host epithelial cells via a zipper-type entry mechanism mediated by the bacterial internalins InlA and InlB. In order to reach the cytosolic replicatory niche, lytic LLO is secreted and ABM provides means of locomotion. Evading host detection and maintaining a permissive environment is controlled by several bacterial effector molecules including InlC, InlK and LntA. Adapted from (Cossart and Lebreton 2014).

Several mechanisms for persistence within the replicatory niche have recently been uncovered. Down-regulation of the host pro-inflammatory response is dependent on secretion of InlC, a virulence factor of the internalin family which prevents phosphorylation of NF-κB and thus prevents nuclear translocation. Both InlK and ActA are associated with ABM and are simultaneously involved in autophagy evasion. Finally, mechanisms for several epigenetic host modifications have been demonstrated. Both LLO action and Met-binding during invasion, trigger histone modifications and upon entry, the virulence factor LntA localizes to the nucleus where it interacts with the newly characterized chromatin component BAHD1. While illustrating that *L. monocytogenes* is capable of reprogramming host gene expression, the exact implications of this capability have yet to be elucidated.

In an extracellular setting, haemolysins serve to rupture erythrocytes in order to generate free iron, a limiting growth factor. Furthermore the nonspecific immune system has to be evaded and haemolysins have also been shown to be cytotoxic towards leukocytes. Additionally, expression of bacterial superoxide dismutase mitigates the effect of free superoxide radicals which play an important role in killing phagocytized bacteria.

Epidemiology. Incidence of listeriosis has initially been increasing since its recognition as food-borne disease but effects of awareness and diagnostic methods are unclear. While typically long incubation periods do pose difficulties for clinical diagnosis, the number of susceptible individuals is on the rise and certain aspects modern processing and handling of foods may be beneficial for growth of *L. Monocytogenes*. Disease rates of 2–15 cases per million population per year have been reported and listeriosis is among the leading case of lethal food-borne pathogen infections. Most recent data, however suggests, that incidence is decreasing again.

Due to its non-fastidious lifestyle, *L. Monocytogenes* has been isolated from a wire array of ecological niches, including soil, sewage and water (both fresh water bodies and estuaries). High persistence (up to 4 years) in soil samples is problematic when contaminated manure is used as fertilizer and biofilm formation poses challenges for eradication from food processing plants. Additionally, the ability to grow in refrigerated foods and resistance towards heat treatment such as pasteurization warrant alertness and special preventive care. Many different foods have been implicated in listeriosis outbreaks, including vegetables (potatoes, radishes and celery), seafood (shrimp, crabmeat and smoked fish), diary products (soft cheese, pasteurized and unpasteurized milk) and meats (poultry, various types of sausages and pâté).

Despite high prevalence in food (studies have found 20–80% of meat product samples and 1–10% of diary product samples contaminated with *L. Monocytogenes*), comparatively few successful infections occur. The bacteria are ingested frequently in small doses and stool sample examinations suggest that 10-70% of investigated populations could be intestinal carriers. While the minimum infectious dose has not been settled definitively, approximations range from 10^6 to 10^9 bacteria.

2.2.4 *Salmonella* Typhimurium

Salmonella are non-sporulating, Gram-negative bacilli, belonging to the family Enterobacteriaceae. The motile bacteria are able to produce peritrichous flagella and diameters span 0.7 to 1.5 µm while typical lengths range from 2 to 5 µm. They are closely related to the genus *Escherichia*, showing only 15% chromosomal sequence disparity. Currently, two distinct species, *S. bongori* and *S. enterica*, within the genus *Salmonella* are recognized, both of which are pathogenic towards a wide array of hosts. *S. enterica* is further divided into 6 subspecies, the most relevant of which for human and domestic animal hosts being *enterica*. A large number of serovars (more than 2500) for *S. enterica* subsp. *enterica* have been characterized and due to an originally mistaken classification as separate species, some serovars are designated with shortened names. *S.* Typhimurium,

therefore is shorthand for *S. enterica* subsp. *enterica*, serovar Typhimurium and to emphasize that Typhimurium is not a species description it is not italicized.

The first description of the genus *Salmonella* dates back to an investigation into swine fever led by Salmon and Smith in 1885. The newly isolated bacterium was wrongly proposed as the etiological agent, as the disease later tuned out to be caused by a virus (Fàbrega and Vila 2013; Haraga, Ohlson, and Miller 2008).

Diseases. Two distinct disease patterns are associated with *Salmonella* spp. infections, typhoid fever and salmonellosis. While the former is exclusively caused by the serovars Typhi and Paratyphi, the latter is associated with several serovars, the most frequent being Enteritidis and Typhimurium, accounting for 65% and 12% of cases worldwide. The current and following sections will not discuss typhoid fever.

Salmonellosis is a diarrheal disease with a short incubation period of 6–24 h, followed by nausea, vomiting, loose or liquid bowel movements, abdominal cramps and fever. Clinical features are similar to those of dysentry and other gastroenteric disease and can include bloody and mucosal stool. In most cases, the infection is self-limiting and symptoms fade away within 4 to 7 days. The most common complication is bacteraemia, which presents in 5% of cases and is more likely to develop in children, especially if malnourished, and immunocompromised individuals. Further manifestations of invasive infections include meningitis, osteomyelitis, cholangitis, pneumonia and endocarditis. While mortality in immonocompetent hosts in developed countries is as low as 0.1%, it can increase to 77% for HIV positive patients in undeveloped countries.

Pathogenesis. In order to reach the small intestine, ingested *S.* Typhimurium first have to defy the hostile environment of the stomach. A set of proteins, summarized as acid tolerance response helps mediate the acidic conditions and improves survival rates. The remaining bacteria subsequently end up in the small bowel and target epithelial cells, with preference towards microfold cells (M cells). Flagellar motility enables penetration of intestinal mucus secretions and improves the chance of reaching the intestinal walls where adhesion can be established. Fimbriae are important attachment factors, capable of interaction with host-cell laminin and fibronectin and provide an initial platform for pathogen induced phagocytosis via trigger mechanism (see section 2.1.2).

S. Typhimurium utilize two separate T3SS systems for host-cell colonization, the first of which (T3SS of SPI-1) mediates invasion. In addition to strengthening initial interactions attaching the pathogen to its target, the needle like

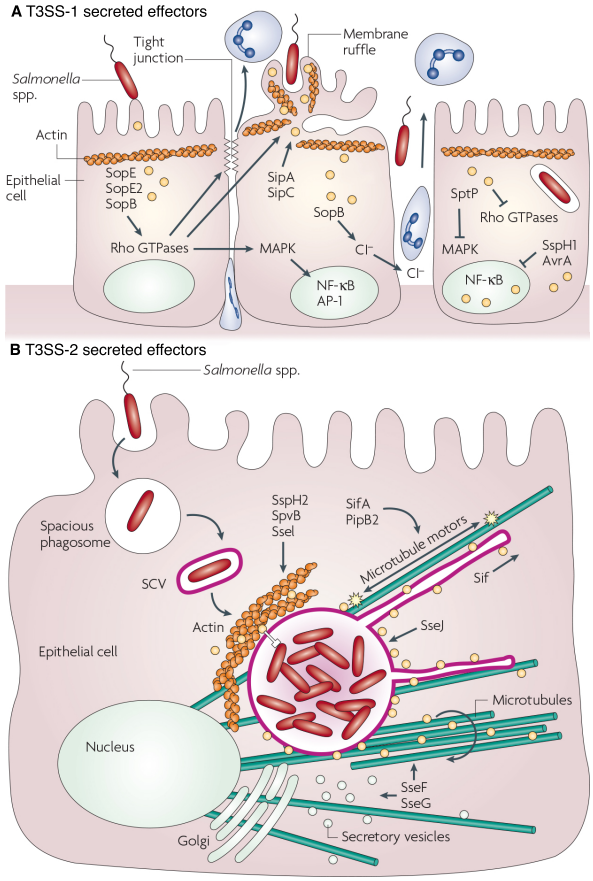


Figure 2.7: Two separate T3SS are encoded in the *Salmonella* genome which serve as central virulence factors at different stages of infection. T3SS of SPI-1 secretes effectors required for triggering internalization (A), while T3SS of SPI-2 is critical for maturation of the phagosome into a replication permissive SCV. Figures adapted from Haraga, Ohlson, and Miller (2008).

structure serves as delivery mechanisms, capable of injecting effector proteins, including SopE, SopE2, SopB and *Salmonella* invasion protein A (SipA). SopE serves as GEF and activates Rho GTPases Rac1 and Cdc42 via GDP/GTP exchange, which in turn initiate cytoskeletal rearrangements. SopE2 is a further GEF, highly homologous to SopE and it is assumed to provide some level of redundancy for SopE as shown by *Salmonella* strains missing the SopE gene.

SopB is a phosphatase, capable of hydrolyzing various substrates, including inositol 1,3,4,5,6-pentakisphosphate, which has been linked to intestinal fluid secretion causing diarrhea and PI(4,5)P_2, involved with membrane rigidity.

The actin binding proteins SipA and SipC stimulates actin polymerization and promotes membrane ruffling, yielding a macropinocytic pocket. Mitogen-activated protein kinases (MAPKs) initiate pro-inflammatory response via NF-κB signaling, leading to the expression of IL-8. Furthermore, damage the tight junctions make the epithelial barrier permissive to bi-directional leakage. After internalization, actin structure is restored and MAPK signaling down regulated by SptP, SspH1 (*Salmonella*-secreted protein H1) and AvrA (*Salmonella* avirulence protein A).

Upon engulfment, maturation of the phagosome and fusion with lysosomes have to be prevented. Unlike other pathogens that escape the digestive mechanisms of phagocytic vesicles by moving to the cytosol, *Salmonella* replicate inside phagosome-derived SCVs. In this setting, the second translocation complex (T3SS of SPI-2) is activated, and used to secrete effectors, including *Salmonella* secretion system apparatus protein B (SsaB) and SifA, capable of interacting with vesicular trafficking mechanisms and guiding the SCV away from its original degradation pathway. Furthermore T3SS-2 translocated effectors such as SspH2, *Salmonella* plasmid virulence B (SpvB) and *Salmonella* secreted effector protein L (SseL) induce actin polymerization events, which relocate the SCV toward a perinuclear position. A last step in creating the intracellular niche needed for replication, is formation of SIFS, extending outwards from the SCV. T3SS-2 secretes effectors such as SifA, PipB2, SseF and SseG, which mediate the microtubule dependent processes by bundling and accumulating filaments and regulating microtubule motor function.

Epidemiology. The global disease burden caused by nontyphoidal salmonellosis is estimated at 90 million cases per year and 150000 deaths. Incidence rates are highest in East and Southeast Asia (up to 4000 cases per 100000 population per year) and both developed and undeveloped countries are affected (incidence rates in Africa are estimated at 320 while estimates for Europe are around 690 cases per 100000 population per year). An estimated 80.3 million or 86% of reported cases are food borne (Majowicz et al. 2010).

In order to control *Salmonella* outbreaks, preventive measures in food production and processing is of major importance. This starts with disease containment in domestic animals, such as vaccination of chickens, enforcing hygiene standards in manufacturing and distribution facilities and ends with proper preparation, exploiting heat sensitivity of the organisms. As the main route of transmission is fecal–oral, good sanitary infrastructure, treatment of

sewage and water processing are crucial prerequisites in combating outbreaks of salmonellosis.

2.2.5 *Shigella flexneri*

Shigellae are small, non-sporeforming, Gram-negative bacilli and belong to the family *Enterobacteriaceae*, along with *Escherichia*, *Salmonella* and *Yersinia*. While flagellar genes are present and their expression is observed under certain conditions, the bacteria are usually described as non-motile and unflagellated. The facultative intracellular parasite shows strong specificity towards human hosts where it typically infects the lower gastrointestinal tract.

Shigella dysenteriae was identified as the etiological agent of dysentery by Shiga in 1897 during an epidemic in Japan with 91000 reported cases and a >20% mortality rate. *S. Flexneri* was first described by Flexner in 1900, while investigating diseases endemic to the Philippines. Recent genetic studies suggest, that *Shigella* spp. belong to the species *Escherichia coli*, rather than forming a distinct genus, as only marginal sequence divergence (1.5%) between *E. coli* and *S. Flexneri* was found (Schroeder and Hilbi 2008; Croxen and Finlay 2010).

Diseases. Bacillary dysentery is an acute infection of the intestine. Mild cases of the disease are self-limiting and afebrile with diarrhea and possibly vomiting as the only symptoms. In as little as 24 hours after onset, bowel movements usually begin to normalize and the condition is resolved within a few days. More severe cases are accompanied by strong abdominal cramps, fever and watery diarrhea containing blood and mucus, indicative of injury to the intestinal epithelium. While still usually self limiting in healthy individuals, recovery takes 10–14 days and relapses are possible. In immunocompromised patients, young children, especially if malnourished, and elderly individuals, life threatening complications including bacteraemia, renal failure, intestinal perforation and toxic megacolon are more frequent. Involvement of the central nervous system and respiratory tract is rare.

Administration of antibiotics is not recommended in mild to moderate cases as the disease can usually be overcome by the immune system and AMR in Shigellae is becoming an increasing concern. Oral rehydration therapy is the most effective treatment, helping the body replenish liquids and salts lost due to diarrhea. For severe infections, the use of antibiotics can become necessary and testing for resistance patterns, if possible, is advised.

Pathogenesis. Main targets of *S. Flexneri* are mucosa of the distal ileum and colon where they enter epithelial cells from the basolateral side. For initial

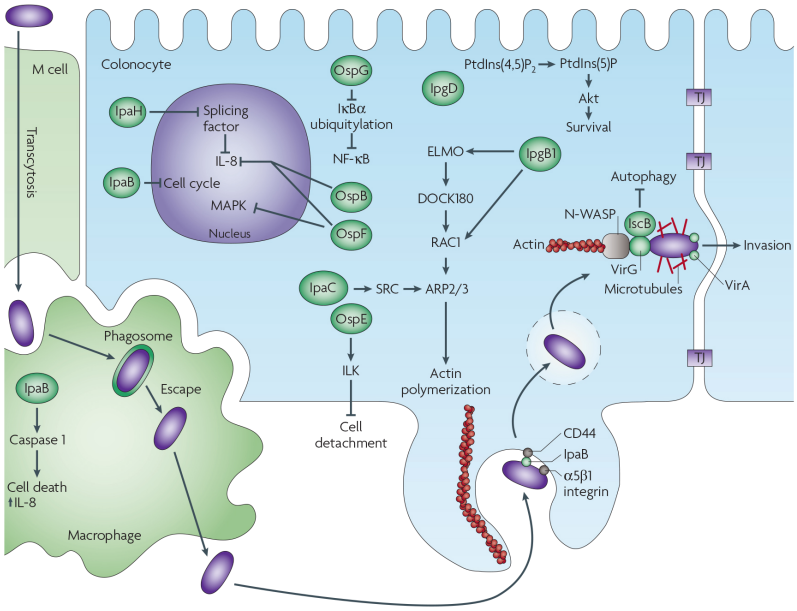


Figure 2.8: *S. flexneri* cross the epithelial barrier of the distal ileum and colon by transcytosis via M cells, followed by escape from macrophage digestion. Internalization into epithelial cells from the basolateral side is mediated by trigger mechanism endocytosis. Several bacterial effectors are instrumental in reaching the replicatory niche and ensuring host survival. Adapted from Croxen and Finlay (2010).

crossing over from the apical side, action of M cells at Peyer's patches is exploited. These specialized enterocytes constantly sample antigens from the gut lumen and pass them to intraepithelially located dendritic cells and lymphocytes. Upon uptake by basolaterally located macrophages, *S. Flexneri* survive digestive action of the phagosomal vacuole by disrupting the surrounding membrane and initiating host-cell apoptosis, mediated by the bacterial effector protein IpaB, capable of acting on a caspase 1 regulated apoptotic pathway (see figure 2.8). The bacteria are subsequently released into the sub-mucosa, where they induce phagocytosis by normal epithelial cells via trigger mechanism (see section 2.1.2).

Initial contact between pathogen and target host cell is mediated by cellular $\alpha_5\beta_1$ integrins and binding of IpaB to CD44 receptors may initiate first actin rearrangements, preparing the cell for uptake. Subsequent injection of at least 6 effector proteins into the epithelial cytosol through the T3SS triggers engulf-

ment of the bacterium in an actin dependent process, involving the small GTPases Rac1 and Cdc42, which recruit the actin nucleation complex Arp2/3. IpaC, a component of the translocation complex, is involved in stimulating Rac1 and Cdc42 GTPase activity through an unknown mechanism and the secreted effector proteins *Shigella* invasion plasmid gene B1 (IpgB1), IpgB2, IpgD and IpaA facilitate actin polymerization. IpgD, an inositol phosphate phosphatase, catalyzes the hydrolysis of PI(4,5)P_2 to PI(5)P which promotes disassociation of cytoskeleton and membrane, increasing actin availability and making the plasma membrane more susceptibility to manipulation. The mechanism of action of IpgB2 remains to be resolved, while IpgB1 is assumed to mimic activated RhoG (Ras homology growth-related), a small GTPase, located upstream in the signaling pathway of Rac1. Finally, binding of IpaA to vinculin induces depolymerization of actin, which is assumed to be important for closing of the phagocytic cup.

The resulting phagocytic vacuole has to be escaped before maturation progresses, which is accomplished in a non-acid dependent fashion, by the translocator proteins IpaB, IpaC and IpaD via membrane lysis. With release into the epithelial cytosol, the replicatory niche of *S. Flexneri* is reached. Actin mediated intracellular motility enables intercellular dissemination (see section 2.1.3) and targeting of epithelial tight junctions initiates break-down of the epithelial barrier, providing more pathogens with access to the basolateral side of the gut lining.

ABM, is driven by activity of two bacterial proteins. VirA is secreted at the forward facing end of the rod shaped bacilli, which promotes degradation of tubulin structures and therefore clearing a path through the dense network of microtubules and surface bound IcsA, also referred to as VirG, facilitates actin polymerization at the opposite end. Both Arp2/3 and N-WASP are involved in actin nucleation, the directed nature of which provides the driving force for locomotion.

In order to maintain their intracellular niche, Shigellae have evolved several strategies. Autophagy is inhibited by IscB (see section 2.1.3) and via the previously mentioned phosphatase action of IpgD, cellular Akt proteins are activated which regulate cell survival and inhibit apoptosis. Furthermore, IpaD interacts with cell cycle regulatory protein MAD2L2, mediating cell cycle arrest. Together with OspE (outer *Shigella* protein E) action on integrin linked kinase, downregulating cell detachment, this prevents turnover of epithelial cells. Inflammatory response is muddled by a combination of at least four bacterial effectors. Cytoplasmically acting OspG inhibits NF-κB, while nuclearly located IpaH and OspB reduce IL-8. Adding to that, OspF dephosphorylates MAPKs that are required for transcription of genes of the NF-κB pathway.

Epidemiology. Estimates by the WHO place the disease burden caused by *Shigella* spp. at 80 million annual cases worldwide, leading to 700000 deaths. Developing countries are disproportionately affected, representing 99% of all cases, as are children less than 5 years old, accounting for 70% of cases and 60% of deaths. In developed countries, incidence rates of 1–2 per 100000 population are typical and Shigellae are common causes of Traveler's diarrhea.

The predominant route of transmission is fecal–oral, highlighting the importance of sanitary precautions for infection control. Proper treatment of fecal matter is important for preventing contamination of drinking water and inhibiting spreading by disease vectors such as house flies. During acute phases, diseased individuals excrete pathogens in large numbers and as few as 100–200 organisms are sufficient of causing a new infection.

2.3 Select Viral Pathogens

In addition to the previous 5 bacterial pathogens, 3 viruses were selected for study within the InfectX RTD project by SystemsX. This section shortly describes each pathogen in terms of physical features, pathogenesis, epidemiology and diseases caused in humans. For each section, a chapter of Craighead (2000) serves as basis.

2.3.1 Adenoviruses

The family *Adenoviridae* encompasses 5 genera of non-enveloped, medium sized (90 nm diameter) viruses, capable of infecting a broad range of vertebrate hosts. The capsid is of $T = 25$ icosahedral symmetry, composed of 720 hexons arranged as 240 trimers which form the triangular facets and 12 penton capsomeres located at the vertices. A homotrimeric fiber glycoprotein protrudes from each vertex, attached to the penton base via interactions of its N-terminal domains and ending in a globular, C-terminal knob. The genome is present as double stranded DNA (Baltimore group I), is non-segmented, linear, 35–35 kb long and encodes 40 proteins.

Adenoviruses were first isolated from human adenoid tissue cultures by Rowe in 1953 and their study led to the discovery of alternative splicing in 1977, a commonly encountered phenomenon among eukaryotes. Currently, 57 serovars are recognized as pathogenic towards humans, all belong to the genus *Mastadenovirus* and are classified into 6 species, labeled A through G. The following sections are mostly concerned with *Human adenovirus C* (Lenaerts, De Clercq, and Naesens 2008).

Diseases. In immunocompetent individuals, adenoviruses seldom cause more than transient disease with many infections even occurring subclinically and fatal outcome being very uncommon. Symptomatic cases usually manifest as respiratory tract infections or conjunctivitis and less frequently as hemorrhagic cystitis, nephritis or gastroenteritis. Infections of the oropharynx can spread to the lower respiratory tract, causing bronchitis, bronchiolitis or pneumonia, which can become chronic, leading to desquamated epithelial tissue and long-term damage to the respiratory mucosa. Heart failure and central nervous system involvement can occur in severe cases. Ocular infections range from mild, short-term follicular conjunctivitis to highly contagious keratoconjunctivitis with possible long-term damage to the cornea. *Human adenovirus C* serotypes are mostly associated with respiratory diseases but have also been implicated in eye infections.

Invasive forms of disease are mostly limited to immunocompromised individuals, including transplant recipients, HIV infected, hereditary immunodeficient and cancer patients subject to chemotherapy. In addition to the lungs, a wide variety of organs has been reported to be affected, such as partoid glands, liver, gall bladder, colon, brain and kidney and pathological changes range from perivascular cuffing to parenchymal necrosis.

Pathogenesis. Initial attachment of virions is mediated by interactions between the globular knobs of viral fiber proteins and target cell CARs (coxsackievirus and adenovirus receptors). In *Human adenovirus C* infections, cellular heparan sulfate proteoglycans serve as additional attachment factors, reinforcing adhesion. Subsequent binding of penton bases to α,β-integrin receptors induces clathrin-mediated endocytosis and leads to loss of viral fiber proteins (see figure 2.9, A).

The adenoviral replication cycle is divided into early (E) and late stages (L), with each seeing expression of a typical set of genes. Upon engulfment by the host cell and triggered by endosomal acidification, hexon bound protein VI disassociates from the capsid structure and induces lysis of the endosomal membrane. The remainder of the now cytosolic virion is shuttled to the nucleus by microtubular transport where viral protein IX recruits kinesin thereby procuring capsid disruption.

Nuclear penetration is mediated by core protein VII and occurs at nuclear pore complexes, leading to transcription of early viral genes by host RNA polymerase II. The resulting proteins manipulate various cellular processes, such as evasion of host immune response (E3, E19), cell cycle (E1A), apoptosis (E1B and E4), autophagy (E1B) and messenger RNA (mRNA) transport (E4), while also promoting viral DNA replication (E2). Modulation if the adaptive im-

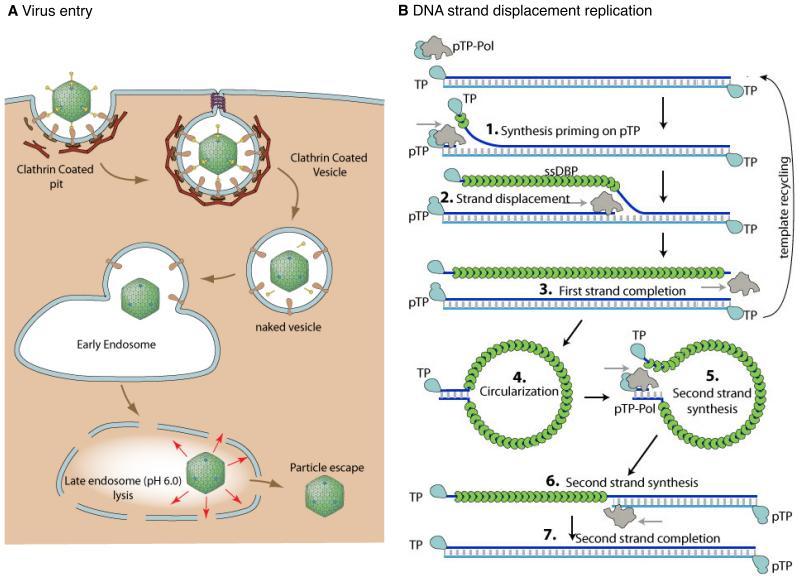


Figure 2.9: Schematic representation of molecular mechanisms for cell entry and genome replication in adenovirus infection. Formation of the endocytic vesicle is mediated by viral fiber proteins interacting with cellular receptors that in turn recruit clathrin. The CCP is pinched off by membrane scission proteins dynamin-1 and dynamin-2, leading to release of a cytosolic CCV (A). The adenoviral genome is replicated via a mechanism described as DNA strand displacement replication whereby single stranded DNA is synthesized from the viral template in a protein primed fashion, which in turn is copied into double strand DNA. Figures adapted from Hulo et al. (2011).

mune response is mediated by the tapasin (TAP) inhibitor E19, as binding of TAP prevents loading of peptides onto major histocompatibility complex class I molecules for display on the cell surface. In order to create an optimal environment for replication, viral protein E1A induces a G1/S cell cycle transition which increases the concentration of cellular enzymes involved in DNA replication. This, however, also leads to accumulation of cellular p53, participating in apoptosis signaling. Several viral proteins, including E1B 55K and E4 orf6 have been shown to inhibit the apoptotic pathway. E1B 19K is involved in activation of autophagy thereby contributing to the delay or inhibition of apoptosis.

A virally encoded DNA polymerase replicates the genome by DNA strand displacement in an unusual protein primed fashion involving precursor terminal protein acting as primer and DNA-binding protein, stabilizing single stranded DNA, as well as several host proteins (figure 2.9, B). With onset of replication, translation of late genes by host RNA polymerase II is initiated, leading to

the production of structural proteins and proteins required for virion assembly. Encapsidation occurs in the nucleus and progeny virions are released by lysis of the host cell.

Epidemiology. Adenoviruses are endemic and ubiquitous, globally causing an estimated 2–5% of all respiratory infections. Distribution is worldwide and incidence is higher in the first half of the year. Children are frequently infected as serological studies show that by the age of 5 years some 50% present antibodies towards the most common species, including *Human adenovirus C*. On the order of 1 in every 10^7 lymphoid cells in the oropharynx harbor fully assembled latent sate virions, making most humans latent carriers. Transmission can both occur via respiratory droplet or fecal-oral routes and virions are very stable towards chemical and physical agents, surviving for long periods outside their host.

Adenovirus outbreaks are a common cause of pneumonia in militaries, leading to the development of live, oral vaccines in the 1960's by the United States Army. The only supplier, however, ceased production and as of 1999, vaccinations could no longer be administered, resulting in reemerging incidence. In the first 5 years after loss of the vaccine, 110000 cases of febrile respiratory illness were reported, of which an estimated 90% are considered preventable. By October 2011, new vaccine again was available and and has been administered to new recruits since.

2.3.2 Rhinoviruses

Investigations into the etiological agent of the common cold within the British Common Cold Research Unit led to the discovery of rhinoviruses in 1953. Initially classified as a separate genus of the family *Picornaviridae*, recent genomic evidence led to a revised taxonomy, recognizing three species of rhinoviruses (A through C) as members of the genus *Enterovirus*. Over 100 distinct serotypes have been isolated from humans, 74 belong to species A, 25 to B and the newly identified species C, currently under active study, may encompass an additional 55 types.

Rhinovirus virions are small (30 nm in diameter), non-enveloped, with a pseudo $T = 3$ icosahedral capsid, consisting of 4 different polypeptides (viral proteins VP1, VP2, VP3 and VP4) surrounding the RNA genome. There are 60 copies of each structural protein and VP1–3 face towards the outside and are responsible for antigenic diversity, while VP4 faces inwards and anchors the RNA core to the capsid. A canyon formed by VP1 and VP3 serves as receptor binding site. The viral genome consists of monopartite, linear, single stranded, positive sense RNA, roughly 7.2 kb in length and encodes a single polyprotein, which

cleaved by viral proteases yields 11 proteins. Instead of a methylated 5′ cap, the RNA genome is terminated by a viral protein (VPg) at its 5′ end (Jacobs et al. 2013).

Diseases. Over half and up to two thirds of all cold-like illnesses are thought to be caused by rhinoviruses. In addition, asymptomatic infections, especially in children are very common with rates among children under 4 years ranging from 12 to 32%. These surprisingly high numbers may to some extent result from virus persistence in hosts that have recovered in addition to disease that has not broken out. In adults, rates of asymptomatic carriage are significantly lower, reported at 0–2%.

In immunocompetent individuals, symptomatic disease typically manifests as upper respiratory infection with clinical syndromes associated with common cold, including rhinorrhea, nasal congestion, sore throat, headache and possibly fever and general malaise. Disease is self-limited, incubation periods are between 12 and 72 hours and within 7 to 14 days symptoms wear off. A common complication is acute otitis media, which has been reported to happen in up to 30% of cases in early childhood. In 41–45% of investigated middle ear infections, rhinoviruses were detected. Further cavities that are frequently infected are the paranasal sinuses. Nose blowing has been suggested as mechanism for spreading the virus and causing rhinosinusitis.

Initially thought to primarily cause benign upper respiratory infections, recent data clearly implicates rhinoviruses in more severe lower respiratory infections. Croup, bronchiolitis and community-acquired pneumonia have been associated with rhinovirus infections and studies have shown that in 12–26% of cases, rhinoviruses were present. Furthermore, a study of children admitted to intensive care units with lower respiratory tract infections detected no other pathogens in 49% of the patients. Rhinovirus species C is implicated more often in severe infections than species A and B. Immunocompromised individuals are predisposed to contracting more serious forms of disease, with morbidity and mortality comparable to that of pandemic H1N1 influenza.

While not typically associated with cytopathogenic effects on epithelia of the upper respiratory tract, cell damage to lung tissue, especially among children, has been documented. Thus, rhinoviruses are linked to exacerbations of chronic pulmonary diseases such as asthma, chronic obstructive pulmonary disease and cystic fibrosis.

Pathogenesis. Members of rhinovirus species A and B are divided into two group according to their host cell receptors. Members of the major group form interactions with ICAM-1 while minor group types (including serotype 1a) as-

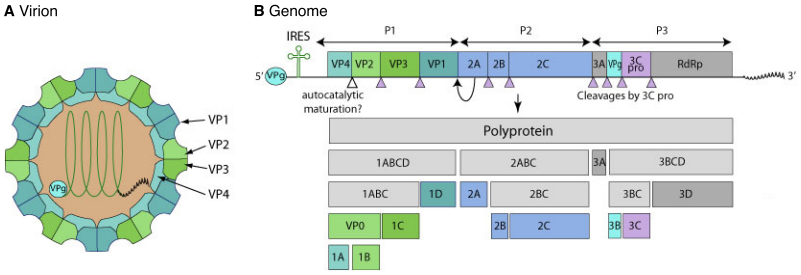


Figure 2.10: Capsid proteins VP1 through VP4 form a pseudo $T = 3$ icosahedral coat, roughly 30 nm in diameter around the RNA genome (A), which is monopartite, linear, 7.2 kb long and encodes 11 proteins (B). Figures adapted from Hulo et al. (2011).

sociate with very low-density lipoprotein receptors. Attachment of species C has very recently been identified as induced by cadherin-related family member 3. Upon receptor mediated endocytosis, pH dependent conformational changes in capsid structure exposes VP4 which has pore-forming properties and facilitates release of viral genomic material into the cytosol.

Host cell ribosomes readily translate the released positive-sense RNA into polyprotein, which is cleaved in *cis* by 2A and 3Cpro, yielding preproteins P1, P2 and P3 (see figure 2.10. P1 is digested into structural capsid proteins while P2 and P3 are further processed to produce the replication machinery. Viral RNA-dependent RNA polymerase synthesizes a complementary, negative-sense RNA strand, primed by VPg, which in turn serves as template for many copies of the viral genome. These can be both translated into more protein and in a later stage of replication also be packaged into progeny virions. Final cell export is mediated by host-cell membrane lysis.

Epidemiology. Despite most infections with rhinoviruses only resulting in mild disease, economic impact both due to health care costs and loss of productivity is considerable. This is primarily owed to the overwhelming prevalence of these pathogens. Respiratory illnesses attributed to rhinoviral infections occur throughout the world and all year round with peak incidences in early fall and spring. Vaccination efforts so far have been futile, mainly because of the large number of serotypes and lack of individual epidemiological data.

Due to acid-sensitivity, fecal-oral infection is unlikely most person-to-person transmission occurs through aerosols and contact spread either direct of via fomites. Extra-host survival times of hours to days have been reported for indoor environments and up to 2 hours on undisturbed skin.

2.3.3 Vaccinia

Vaccinia virus is a species within the genus *Orthopoxvirus*, alongside the exceptionally virulent *Variola virus*, the etiological agent of smallpox. Immunological similarities between the two species allows for cross-protection, which coupled with the large discrepancy in pathogenicity presents a fortunate opportunity for artificially inducing acquired immunity. This was recognized by Jenner in 1798, while investigating the zoonosis of *Cowpox virus* and subsequent susceptibility towards contraction of smallpox. The origins of vaccinia are unknown. It has been speculated to have derived from cowpox or smallpox, to be a hybrid of both and of being the prototype orthopoxvirus, as well as descending from an extinct ancestor.

The virions are large and brick shaped, measuring 200 by 250 by 340 nm and exist as both mature virion (MV) and extracellular virion (EV). Structurally they are unusually complex. The linear, double-stranded DNA genome, roughly 200 kb long, is encased in an S-shaped, tube-like nucleocapsid with an outside diameter of 50 nm, a cavity of 10 nm and an overall length of 250 nm. Additionally, 47 viral proteins, of which 16 are involved in early mRNA synthesis and 28 have no known enzymatic function, are packed into a core structure. The core wall consists of two layers, the palisade (outer) layer which is 17 nm thick and an inner smooth layer, measuring 8 nm across. Centered on the top and bottom faces, two proteinaceous lateral bodies separate core from the surface membrane, forming the virion core into a biconcave disc with dumbbell shaped vertical cross sections. The lipidic surface membrane also consist of two layers, the outermost measuring 9 nm and the innermost domain is 5 nm thick. EV virions are additionally wrapped in a membrane derived from Golgi cisternae (Marennikova, Condit, and Moyer 2005; Condit, Moussatche, and Traktman 2006).

Diseases. While infection with variola causes severe disease manifesting in skin lesions covering the whole body, accompanied by 20–40% mortality rates, the closely related *Vaccinia virus* is far less invasive. Virulence of the latter pathogen is so low that it has been routinely used as live vaccine against the former. Owing to the massive effort put into the worldwide fight against smallpox led by the WHO in the late 1960's and throughout the 1970's, the disease was found to be eliminated by 1980. At the center of the smallpox eradication program was the administration of freeze-dried, calf lymph derived vaccinia with a bifurcated needle, by multiple pricking of the skin. Towards the end of the initiative, 200 million vaccinations were performed annually.

The predominant reaction to vaccinia inoculation is localized, self-limited disease. After an incubation period of of 3–4 days, a papule with a sunken center

develops at the site of infection, accompanied by erythema and swelling. Over the following days the papule increases in size and liquid begins to accumulate within. Fever may develop around days 7–10, possibly followed by malaise, headache and anorexia. Local lymphadenopathy is frequently encountered and days 8–10 typically mark the beginning of involution of the papule, which subsequently drys out and forms a scab.

Of great concern for routine vaccination procedures are complications which inevitably occur in a small numbers. Atypical side effects develop in roughly 1 per mill of cases and potentially life threatening reactions usually manifest as neurological (477.4 cases and 47.0 fatalities per 1 million) or cutaneous disease (278.4 cases and 0.2 fatalities per 1 million). Predisposing conditions for central nervous system involvement are not known and despite decades of inquiry into this pathology, it remains poorly understood. Symptoms range from febrile seizures to severe encephalitis, but so far no neuropathological characteristics have been identified. Complications affecting the skin and mucous membranes are classified as progressive vaccinia, eczema vaccinatum and generalized vaccinia and disease severity decreases in this order. Predisposing conditions for progressive and generalized vaccinia are immunodeficiencies while a history of eczema is a risk factor for eczema vaccinatum.

Pathogenesis. Initial attachement is mediated by interaction between viral proteins and cellular heparan sulfate chains. For cell entry, various strategies have been reported, dependent on the virus strain. WR strains induce macropinocytosis and proteins A25/A26 act as fusion suppressors that only cease action under acidic conditions encountered in maturing endosomes, while other strains, such as Copenhagen, present no A25/A26 on their outer membrane and fuse directly with the target. Due to the additional membrane of EVs, an alternative entry mechanism is required. In a currently not well understood fashion, the outer membrane is lost by non-fusogenic disruption, followed by fusion of the inner virion membrane. All pathways lead to cytosolic localization of virions devoid of their envelope (see figure 2.11).

Members of the *Poxviridae* family are special among Baltimore group I viruses in that their genome encodes all necessary replicatory proteins, allowing for cytosolic localization. Replication is temporally tightly regulated and consists of distinct phases (early, intermediate and late gene expression). Each class of genes encodes factors capable of initiating the succeeding stage, providing transcription level regulation. Uncoating of the core structure releases early proteins, including RNA polymerases and enzymes for mRNA processing which start transcribing early genes. At least 50 different products, such as DNA replicatory enzymes, additional RNA polymerase, mRNA processing machinery, host defense molecules and intermediate gene transcription factors, have

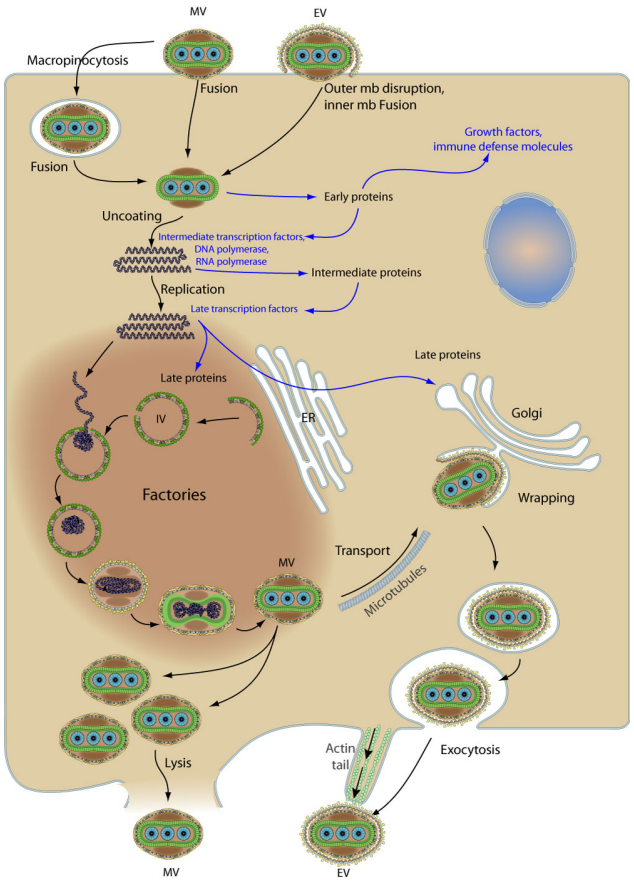


Figure 2.11: Replication cycle of *Vaccinia viruses* for both mature virion (MV) and extracellular virion (EV). Cell entry is either macropinocytic or proceeds via membrane fusion, followed by uncoating and replication. Host exit proceeds by either lytic or exocytic mechanisms. Figure adapted from Hulo et al. (2011).

been identified and account for 25% of the viral coding capacity. Early gene transcripts are detectable 20 minutes after cell entry and reach their productive peak within 100 minutes of infection.

Expression of intermediate genes is initiated by accumulation of intermediate transcription factors and the onset of DNA replication. Only 7 products of this

phase are known, which functionally are mostly concerned with host defense, DNA/RNA metabolism and commencement of the final phase. Beginning 100 minutes after infection, intermediate gene transcription reaches its peak at 120 minutes and is thereafter superseded by late gene transcription, beginning 140 minutes after cell entry. Products of the final phase comprise a large number of genes (up to 75% of the vaccinia genome) and include enzymes needed for initiating replication (RNA polymerase and modification proteins), early transcription factors and structural proteins, as well as virion assembly machinery.

DNA replication not only serves for progeny virions, but also to increase the concentration of templates used for gene expression. Both DNA synthesis and virion assembly occur within factories, readily visualized by electron microscopy as electron dense cytoplasmic inclusion bodies. Owing to the complex virion structure DNA packaging and virion assembly is an involved procedure with is incompletely understood.

Epidemiology. It is unknown if a natural reservoir of vaccinia exists. It has been speculated that the virus is maintained only within research laboratories and vaccination production facilities, while others have implicated some rodent species as possible reservoir hosts. Small scale zoonotic outbreaks of vaccinia have been documented in Brazil and it was initially suspected that these were linked to vaccine that had escaped into the environment. Recent phylogenetic studies however were able to rule out this explanation but were unable to shed further light into underlying epidemiological mechanisms.

While transmission from vaccinees to unvaccinated individuals is rare, direct contact transmission is possible and such occurrences have been documented. Special care has to be taken to avoid direct contact between recently vaccinated and individuals predisposed towards developing complications.

2.4 RNA Interference

First described only two decades ago, regulation of gene expression by short strands of RNA has become an indispensable tool to both experimental biology and bioinformatics. Recognizing the importance of applications made possible by this discovery, the 2006 Nobel prize in Physiology or Medicine was awarded to Fire and Mello who studied RNA interference in the nematode worm *Caenorhabditis elegans* and published their findings in 1998. Building on studies by Guo and Kemphues, who showed that sense RNA, as well as antisense RNA was capable of suppressing gene expression, Fire, Mello and coworkers found that double stranded RNA was at least ten-fold more effective as silencing agent than individual single stranded fragments. Further investi-

gations showed that several gene regulatory processes, previously thought to be unrelated, were in fact manifestations of RNA interference and that the underlying mechanism was conserved in many, if not most, eukaryotic organisms (Hannon 2002).

The RNA interference (RNAi) pathway can take as input two separate types of RNA molecules, micro RNA (miRNA) and siRNA, of differing origins. While miRNAs are endogenous and purposively employed in post-transcriptional regulation of gene expression, siRNAs are exogenous synthetic or viral inducers of gene suppression, in which case, RNA interference can be viewed as an immune response to foreign genetic material. Parsimony-based phylogenetic analysis of involved genes suggests that the key components to an RNAi system were already present in the last common ancestor of eukaryotes and were subsequently lost or extensively simplified in some protists. The original function of RNAi is hypothesized to be that of a defense mechanism against genomic parasites as indicated by the extent of its conservation, whereas miRNA-directed silencing most probably was introduced at a later point in evolution (Cerutti and Casas-Mollano 2006).

2.4.1 Molecular Mechanism[3]

RNA interference refers to three separate mechanisms for regulation of gene expression by small RNAs, as visualized by figure 2.12. While siRNAs are involved both in transcriptional and post-transcriptional gene silencing, the miRNA pathway is focused on translational repression. The severity of action on the targeted genes once again highlights the differing purposes of the siRNA and miRNA pathways, one tasked with inhibitory and the other with regulatory measures.

Translational repression by miRNA. The biogenesis of miRNA occurs in the nucleus and is initiated by RNA polymerase II transcription of long (>1000 nt) primary miRNA (pri-miRNA) segments, characterized by double-stranded hairpin loops with single-stranded 5′- and 3′-terminal overhangs which are polyadenylated and capped. Subsequent processing by the microprocessor complex consisting of the RNase (ribonuclease) III family enzyme Drosha and DGCR8 (DiGeorge syndrome critical region gene 8) yields ~60–70 nt stem-loop structured precursor miRNA fragments. DGCR8 recognizes pri-miRNAs by the junction of stem and single-stranded overhang and helps positioning the substrate for endonucleolytic cleavage by Drosha at a site ~11 nt from the junction.

[3]This section is based on the three review articles by Wilson and Doudna (2013), Kim and Rossi (2007) and Carthew and Sontheimer (2009).

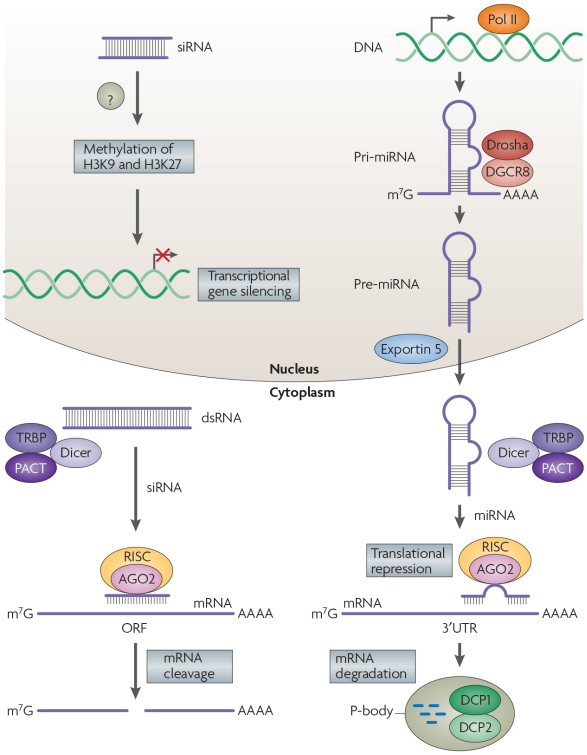


Figure 2.12: RNA interference comprises of three distinct mechanisms that yield control over gene expression. Exogenous double-stranded RNA are processed into siRNA fragments that both act inside the nucleus as transcriptional silencing agents and in the cytoplasm, post-transcriptionally cleaving mRNA strands. Endogenous miRNA is synthesized by RNA polymerase, originates from the nucleus in processed form and mediates milder translational repression Figure adapted from Kim and Rossi (2007).

Nuclear export is mediated by the transport facilitator exportin-5 and is Ran-GTP dependent.

In the cytoplasm, pri-miRNAs are targeted by Dicer and the associated double stranded RNA (dsRNA) binding proteins TRBP (TAR RNA-binding protein) and PACT (protein activator of protein kinase PKR). These process their substrate into 21–25 nt dsRNA strands with 2 nt overhangs at the 3′ termini and phosphate groups at each of the recessed 5′ ends. The mature miRNAs are loaded onto Ago (Argonaute protein family) by Dicer, which leads to the formation of RNA-induced silencing complex (RISC). Concomitantly with RISC-

loading, one of the two RNA stands is selected as guide strand whereas its complement (the passenger strand) is ejected and degraded. Thermodynamic asymmetry between the two strands is exploited in this step and the strand with the less stable 5′ end is preferred. As opposed to strand separation in siRNAs, the passenger strand is not cleaved but rather unwound by helicase activity, facilitated by imperfect sequence alignment.

Finally, active RISC, exposing the Ago-bound guide strand, interacts with the 3′ untranslated region of mRNA targets and directs translational repression and mRNA degradation. Sequence homology between mRNA and the miRNA seed sequence (the first 2–6 or 2–8 nt from the 5′ end) is critical while mismatched nucleotides towards the 3′ end of the miRNA are readily tolerated. The extent of base-pairing influences the subsequent mechanism of silencing, ranging from direct target cleavage (perfect match) over deadenylation (followed by degradation) to nonendonucleolytic translational repression (imperfect match).

Post-transcriptional gene silencing by siRNA. Precursors to siRNA are long, linear, perfectly base-paired double stranded sequences of RNA, typically of exogenous origin either introduced directly into the cytoplasm, or taken up from the environment. A complex consisting of Dicer and several RNA-binding proteins are responsible for trimming down dsRNA fragments to the appropriate size for loading onto Ago2. Of the four Ago family members in humans, capable of associating with miRNA, only Ago2 seems to be involved with siRNA. Furthermore, Ago2 is the only mammalian Argonaute protein family member bearing endonucleolytic functionality and therefore capable of directly cleaving targeted mRNA.

Strand selection is again based on differences in stability of base-pairing at the 5′ termini with the weaker interacting end being favored as guide strand. Accuracy of discrimination can be low, leading to incorporation of both strands with equal frequency. In contrast to miRNA loading, the passenger strand is not merely separated but directly cleaved by Ago2 and the differing treatment seems to only depend on perfect strand complementarity given in siRNA and absent in miRNA. Upon RNA incorporation, RISC is formed and activated by cleavage of the passenger strand. The 3′ guide strand end is bound by the Ago protein's PIWI-Argonaute-Zwille domain, while the 5′ end interacts with Argonaute middle domain, closely located to the catalytic RNase H-like P-element-induced whimpy testes domain.

Post-transcriptional gene silencing is accomplished by endonucleolysis of perfectly matching mRNA precisely at the phosphodiester linkage between bases 10 and 11 relative to the 5′ terminus of the siRNA guide strand. Follow-

ing cleavage, the target disociates, freeing RISC for further catalysis, and the mRNA fragments are degraded by cellular exonucleases. Imperfectly matched mRNA may be targeted, much like it is the case with miRNA, leading to siRNA off-target effects which are of great practical importance.

Transcriptional gene silencing by siRNA. In addition to post transcriptional action of siRNA, nuclear inhibition of gene transcription has been described in many eukaryotes. Diced siRNA fragments are transported into the nucleus where they are assembled with a group of proteins, including Ago1, to form RNA-induced transcriptional silencing complex (RITS). Currently only incompletely understood, the siRNA guide strand is thought to recognize RNA transcripts as the emerge from RNA polymerase II, followed by recruitment of factors that enable covalent modifications of nearby histones. Methylation of lysines 9 and 27 in H3 by histone methyltransferases leads to chromatin compaction and heterochromatin formation. RITS has also been shown to induce direct methylation of DNA, repressing gene expression even further.

Contributing to the potency of RNA interference, engagement of RITS with nascent transcripts activates the RNA-dependent RNA polymerase complex, capable of generating secondary siRNA fragments and therefore amplifying silencing capabilities. The role of this reinforcement mechanism has been firmly established in many eukaryotic RNAi systems with the notable exceptions of vertebrates and insects. Whether a similar system exists in these organisms remains an open question.

2.4.2 Biological Function

The mechanisms of RNA interference have most probably evolved in order to protect against foreign genetic material such as parasitic DNA sequences or viral RNA. Transposable elements (transposons) are DNA sequences that are mobile within the genome, can make up a significant fraction of eukaryotic genomes and are typically considered non-coding. Transposition is mediated by transposases, enzymes often encoded within the transposons themselves, that act on specific sequences at the transposon ends and cause unspecific insertion into new target sites. Retrotransposons move via an RNA intermediate which is reverse transcribed to DNA and inserted, while DNA transposons employ a cut and paste mechanism. Retroviruses therefore can be viewed as transposons and in general, transposable elements are a form of selfish DNA that often incur deleterious effects.

RNAi is an important regulatory force to transposon activity, both by processing transcripts of retrotransposons, thereby reducing their concentration and eliciting epigenetic modifications, as well as transcriptional inhibition via het-

erochromatin formation. The importance of keeping transposable elements in check is highlighted by their prevalence, with roughly half of the human genome being thought to derive thereof.

Antiviral mechanisms are particularly important in organisms lacking an adaptive immune system as found in vertebrates and exploiting the orthogonality of most genomic systems to double-stranded RNA puts RNA interference in a powerful position. Corroborating this notion is the observation that, in *Drosophila melanogaster*, three key proteins of the RNAi pathway (Dicer-2, Ago2 and R2D2, a protein involved in RISC loading) are among the top 3% in terms of genetic instability. Furthermore, miRNA pathway paralogs to these three genes (Dicer-1, Ago1 and R3D1), not being involved in immune response, evolve at a much slower rate (Obbard et al. 2009).

Although small RNA-guided, Ago-dependent up-regulation of gene expression (termed RNA activation or RNAa) has been described, most regulation of gene expression by miRNA is of inhibitory nature. This widespread mechanism, consisting of >1000 miRNA sequences (as much as 5% of the human genome) controls at least 30% of human genes and is responsible for vital processes including cell growth, tissue differentiation and cell proliferation.

2.4.3 Applications

In *C. elegans*, RNA interference is especially powerful, making it a popular model organism for RNAi. Not only is delivery efficiently possible simply by feeding the nematodes with bacteria such as *E. coli* that carry the desired dsRNA, but the resulting gene silencing effects are hereditary. Moreover, RNAi response is not stoichiometric but catalytic, is amplified in a feedback loop and in many organisms, systemic spread has been documented.

The initial burst of excitement surrounding possible applications was somewhat moderated by difficulties in applying RNAi to mammalian systems. At first it seemed impossible to use this technology in somatic cells as the introduction of dsRNA is typically met with overwhelming non-specific responses, including PKR (protein kinase RNA-activated), which leads to arrest of translation and apoptosis. This issue was shown to be overcome by exclusively using siRNAs duplexes of 21–23 nt with 2-nt 3′ overhangs that mimic Dicer products and are too short for inducing PKR. Mammalian RNAi response, however is transient, lacking amplification and spreading mechanisms documented in other organisms (mainly plants and worms) and delivery, especially in vivo remains problematic. A further issue that continues to be an actively researched area of interest is that of off-target effects (OTE), which considerably complicate the interpretation of phenotypic data.

Gene knockdown studies. Large-scale loss-of-function (LOF) and modifier or synthetic lethality screens are readily possible by means of RNAi based high throughput screening (HTS). Such experiments usually proceed by arraying libraries of gene specific siRNAs onto microtiter plates (96 and 384 well formats are common), followed by the addition of liquid cell cultures. After an appropriate transfection time, the cells may be subjected to an additional treatment, such as exposure to drugs or pathogens (modifier screen) or LOF phenotypes can be investigated directly. Assay readout is performed via optical measurements such as detection of fluorescence or luminescence signals or by microscopic imaging (confocal or wide-field).

Transcriptional reporters, fluorescent dyes that detect enzymatic activity and protein-modification specific antibodies have been employed in plate reader-based investigations which yield a single numerical readout per well. This quantitative approach is contrasted with microscopy based assay read-outs that are able to capture spatial information on antibody stained proteins, fluorescently labeled cellular structures and green fluorescent protein (GFP) expression, yielding much more data per well. Significant challenges incurred by automated high-content imaging have successfully been addressed by computational image analysis software.

A multitude of technical and biological noise sources contribute to serious problems in interpretability and comparability of observed data. Common to all HTS approaches, errors arise from difficulties guaranteeing equal conditions in a large number of parallel experiments. Liquid dispensing errors, temporal disparities caused by bottleneck resources such as imaging equipment and spatial discrepancies, for example inhomogeneous temperature distribution over the plate or liquid evaporation in border wells, are only a few issues that come to mind. Biological sources of error include OTEs, varying potency of reagents (both the knockdown strength and time required to achieve optimal knockdown) and obscuration of assay phenotype by knockdown phenotype (e.g. cell death). Furthermore, incorrect gene models lead to errors in library design and detection may be hampered by weak phenotypes. Replicates, although expensive in large-scale experiments and control wells embedded in every assay plate are indispensable measures in order to assess reproducibility of the data (Echeverri and Perrimon 2006; Perrimon and Mathey-Prevot 2007).

Despite being a young technology, RNA interference has already proven itself as an invaluable tool and has yielded many insights with significant impact on various fields. A review by Mohr, Bakal, and Perrimon (2010) lists some of these findings which lead to refined understanding of cell proliferations, cancer biology, cell cycle regulation, mitochondrial diseases, signal transduction, RNA biology and pathogen response.

Table 2.3: A non exhaustive list of RNAi based drugs that currently are in clinical trials. The data was obtained from the clinicaltrials.gov database (McCray and Ide 2000).

Company	Disease	Delivery system	Status
Alnylam	Transthyretin-Mediated Amyloidosis	siRNA–GalNAc conjugate	Phase III recruiting
Alnylam	Antitrypsin Deficiency Liver Disease	Liposome	Phase II recruiting
Alnylam	Acute Intermittent Porphyria	siRNA–GalNAc conjugate	Phase I recruiting
Silenseed	Advanced pancreatic cancer	Polymer (LODER)	Phase III planned
Sylentis	Dry eye syndrome	Naked siRNA	Phase II completed
Sylentis	Open angle glaucoma	Naked siRNA	Phase II recruiting
Tacere	Chronic hepatitis C	Adeno-associated virus vector	Phase II recruiting
Tekmira	Advanced Hepatocellular Carcinoma	Liposome	Phase II recruiting

Biotechnological applications. Intercellular, systemic spread of RNAi response in plants and even its heredity over several generations have been documented and it comes as no surprise that the technology is investigated for possible utilization in crop improvement. Removal of plant endotoxins by targeting genes of toxin biosynthesis has been accomplished, leading to the production of decaffeinated coffee plants (knockdown of theobromine synthase), tobacco with reduced concentration of carcinogenic compounds (inhibition of nicotine demethylase activity) and edible cotton seeds (reduction of delta-cadinene synthase leads to low levels of gossypol, a toxic terpenoid), which are naturally rich in dietary protein.

In addition to investigations with consumer health in mind, improvements in environmental resistance have been studied in many organisms. Susceptibility to bacterial and viral pathogen infection has been reduced in rice, bean, barley and lemon, while fungal resistance has been increased in potato, tobacco and wheat. Successful RNAi application as insecticide has been shown in cotton and maize and even improved resistance to parasitic weeds could be demonstrated in transgenic tomato plants. Despite these achievements, concerns over environmental issues and adverse health effects have so far prevented RNAi based genetic modifications from exiting experimental stages (Saurabh, Vidyarthi, and Prasad 2014).

Therapeutic potential. Great promise lies in therapeutic application of RNAi as theoretically, any gene can be targeted, yielding unparalleled flexibility not encountered with typical small molecule drugs. An initial obstacle to harnessing this power in humans is the issue of delivery. Systemic spread of naked siRNAs is hampered by kidney filtration, phagocytic uptake and degradation by serum RNases. Movement across capillary walls is not readily possible for molecules larger than 5 nm and phagocytes patrol the extracellular matrix, ingesting foreign genetic material. Furthermore, polyanionic macromolecules do not easily penetrate hydrophobic cellular membranes.

Topical or local administration offers advantages, including increased bioavailability and reduced side effects and is therefore preferred for treatment of eye, skin and mucosal diseases, as well as localized tumors. If targets are not localized or difficult to reach, injection into the bloodstream may provide a mode of systemic application. Chemical modification of the RNA backbone (2′-O-methylation or 2′-fluorination of ribose) has been shown to provide resistance to RNase and covalent attachment to cholesterol promotes cellular uptake. Encapsulation of siRNA in liposomes and cationic polymers are further proven techniques for improving extracellular stability, stimulate endocytosis and facilitate endosomal escape.

Some sort of target selectivity is desirable on order to avoid high dosages, associated concerns of toxicity and potential OTEs in non-target tissue. Coupling of siRNA reagents to antibodies specific for HIV envelope glycoproteins has been successfully employed for selectively entering infected cells, while aptamers (oligonucleotides specifically engineered for binding a given target), carrying siRNAs have also been shown to be capable homing mechanisms.

Viral delivery of of RNAi inducing agents presents an alternative technique to the above and in case of retroviral transport vessels, short hairpin RNAs (shRNAs) that are reverse transcribed and integrated into the host genome, provide stable expression and prolonged RNAi activity. Adenoviral and adeno-associated virus vectors have also been employed, resulting in a more transient response. While health concerns associated with perpetuity of gene therapy no longer apply, repeated administrations may prove problematic, triggering strong immune response and thereby limiting therapeutic potential.

Currently, multiple siRNA based therapeutics are in clinical trials, including stage III studies (see table 2.3). Due to the unprecedented pace at which RNAi technology went from discovery to development of applications, much uncertainty remains surrounding long term effects of exposure to such drugs. Chronic diseases including hepatitis C or HIV infections require life-long treatments and consequences of repetitively triggering RNAi response has not been thoroughly studied (unwanted changes to chromatin structure for example,

are one area of concern). Apart from safety issues, several implementation aspects require further study. While neurodegenerative diseases have successfully been treated in mouse models via direct injection into the brain, this is not easily feasible in human patients and currently no delivery vehicle capable of crossing the blood-brain barrier. (Kim and Rossi 2007; Whitehead, Langer, and Anderson 2009),

Chapter 3

Data

Tasked with elucidating components of the human infectome for a set of bacterial and viral pathogens, an interdisciplinary consortium of research groups generated kinome- and genome-wide siRNA screens for each of the investigated pathogens. Furthermore, the SystemsX RTD project InfectX encompasses both automated microscopic imaging and computational image analysis and the acquired data is publicly available online at the InfectX website. Particularities of data acquisition are the focus of this chapter, as basic understanding of the experimental setup and analysis pipeline is crucial for further investigations. Much of the information contained in this chapter has been published by Rämö et al. (2014) and is summarized for the reader's convenience.

3.1 InfectX Workflow

Due to the large scale yielding from screening multiple libraries and using different pathogens, while also desiring experimental replication, several labs were involved in carrying out wet-lab procedures. In order to obtain reproducible results, a strong emphasis was put on developing unified work-flows for cell culture, siRNA transfection, pathogen infection and imaging. Figure 3.1 summarizes the central steps, beginning with siRNA libraries arrayed onto 384-well plates that are used for transfection of added cells, carrying on with pathogen infection, subsequent microscopic imaging and concluding with computational image analysis.

The model system chosen for investigation is HeLa (ATCC CCL-2), the oldest and most wide spread human cell line and a proven system for studying

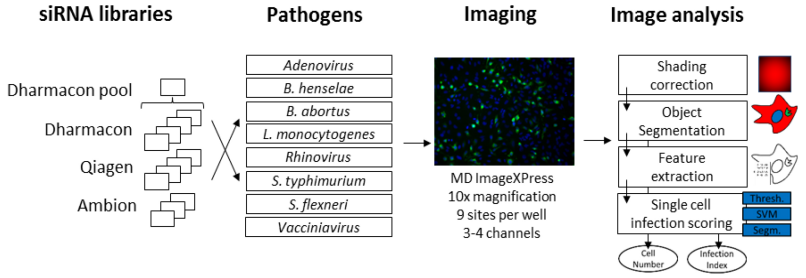


Figure 3.1: A total of 11 single siRNA libraries, pdroduced by 3 separate manufacturers, (4 from Dharmacon, 4 from Quiagen and 3 from Ambion), as well as one pooled library (Dharmacon) were screened with 8 pathogens. Plates were imaged under wide-field microscopes and the resulting images run through an image analysis pipeline. (Rämö et al. 2014).

infectious disease[1]. Culturing was performed at 37 °C, under 5% CO_2 atmosphere for maintaining optimal pH, using Dulbecco modified Eagle medium supplemented with 10% inactivated fetal bovine serum (FBS), both supplied by Invitrogen.

While genome-wide siRNA screens were also produced within the InfectX framework, this report focuses on kinome-wide investigations. Introduced by Manning et al. (2002), the term kinome refers to the subset of genes encoding protein kinases[2]. As phosphorylation reactions have been identified to constitute the most widespread signaling mechanism in eukaryotic cells, the set of

[1]Collected 60 years ago from a cervical adenocarcinoma, these epithelial cells have led to much insight into human cell biology. Prior to their discovery, attempts at growing human tissue in vitro were futile and development of protocols for sustaining cancerous human tissue was thought to hold great promise for cancer research. One of the early successes involving HeLa cells was the development of a polio vaccine. For this endeavor, a large amount of human cells were needed and the installment of a production facility capable of meeting the high demand might have contributed significantly to the predominance of these cells. (Masters 2002)

[2]Kinases are part of the larger enzymatic family of transferases and catalyze phosphorylation reactions of the form

$$\mathrm{A{-}O{-}PO_3{}^{-2} + H{-}X{-}B \rightleftharpoons B{-}X{-}O{-}PO_3{}^{-2} + H{-}O{-}A} \tag{3.1}$$

where A represents the donating (typically ATP) and B the accepting molecule. Kinases are further subdivided according to the type of acceptor which can be an alcohol, carboxy, nitrogenous or phosphate group, or in case of protein kinases, a tyrosine, serine, threonine or histidine residue.

$$\mathrm{A{-}O{-}B + H{-}O{-}PO_3{}^{-2} \rightleftharpoons A{-}O{-}PO_3{}^{-2} + H{-}O{-}B} \tag{3.2}$$

$$\mathrm{A{-}O{-}PO_3{}^{-2} + H{-}OH \rightleftharpoons A{-}OH + H{-}O{-}PO_3{}^{-2}} \tag{3.3}$$

Phosphorylases are a further group of transferases that involve phosphate but unlike kinases, they utilize inorganic phosphate sources (3.2). Often phosphorylases are involved in breaking down biological polymers such as polysaccharides and polynucleotides. Finally, phosphatases catalyze the reverse reaction, the removal of a phosphate group (3.3).

Table 3.1: Number of replicates performed for each of the pathogens and siRNA libraries. The primary values indicate how many were performed in total while the value in parenthesis represents the number of screens that turned out to be unusable and had to be discarded. The effectively available number of replicates is the difference between the two (Rämö et al. 2014).

Pathogen	Dharmacon (1x pooled)	Ambion (3x single)	Quiagen (4x single)	Dharmacon (4x single)
B. abortus	8 (1 rem.)	2	1	2
B. henselae	5 (1 rem.)	4 (1 rem.)	2 (1 rem.)	1
L. monocytogenes	4	4 (2 rem.)	1	1
S. flexneri	6 (2 rem.)	2	1	1
S. typhimurium	7 (4 rem.)	3	1	1
adenovirus	8 (1 rem.)	2	1	1
rhinovirus	8 (2 rem.)	2	1	1
vaccinia virus	2 (1 rem.)	2	1	1

518 genes identified by Manning et al. are a popular target for functional genomics studies. Up to 30% of intracellular proteins may be phosphorylated, leading to 20000 phosphoprotein states, all regulated by expression of varyingly substrate specific kinases (Johnson and Hunter 2005). Due to the importance of kinases for cell behavior, they represent a major drug target in cancer therapeutics and might therefore also be attractive for HDT.

In order to offset potential bias introduced by siRNA design paradigms employed by manufacturers, libraries from three different companies (Abion, Dharmacon and Qiagen) were sourced. To further account for the effect of OTEs, several siRNA sequences per target were used for both pooled and unpooled experiments. The Ambion Silencer Select Human Kinase siRNA Library V4 targets 710 genes of kinome-associated proteins with 3 sequences each, while the Qiagen HP OnGuard Human Kinase siRNA Set V4.1 comprises of 718 targets and 4 siRNAs per gene. The Dharmacon Human ON-TARGETplus siRNA Protein Kinase Libraries are designed with 715 genes in mind and are both available in 1 siRNA (unpooled) or 4 siRNAs per well (pooled) formats.

Depending on library and pathogen, screens were repeated 1–8 times (see table 3.1). The primary values denote the total number of replicated performed and the values in parenthesis indicate the number of screens that had to be removed due to issues with transfection, weak signal intensities or usage of a protocol other than the one eventually agreed upon. The Dharmacon pooled screen was used for optimizing the assay procedures which is why for almost all pathogens some replicates had to be removed. The number of available screens is the difference between primary and parenthesized values.

Table 3.2: Despite putting much emphasis on using identical protocols throughout all screens, some assay parameters were fine-tuned in order to obtain phenotypes such as infection and cell counts that are similar among the investigated pathogens (Rämö et al. 2014).

Pathogen	Seeded cell number	MOI	Pathogen entry time	Total infection time	DNA stain	Actin stain[a]	Pathogen stain	Additional stain[b]
B. abortus	500	10000	4 h	44 h	DAPI	DY-547	GFP	-
B. henselae	300	400	24 h	24 h	DAPI	DY-547	GFP	-
L. monocytogenes	600	25	1 h	5 h	DAPI	DY-647	GFP	Alexa546
S. flexneri	600	15	30 min	3.5 h	Hoechst	DY-495	DsRed	Alexa647
S. typhimurium	550	80	20 min	4 h	DAPI	DY-547	GFP	-
adenovirus	700	0.1	16 h	16 h	DAPI	DY-647	GFP	-
rhinovirus	1000	8	7 h	7 h	DAPI	DY-647	GFP	-
vaccinia virus	600	0.125	1 h/8 h[c]	24 h	Hoechst	DY-647	GFP/RFP[c]	-

[a] Phalloidin-based actin stains were supplied by Dyomics and depending on absorption wavelength, different imaging channels were used: GFP for DY-495, RFP for DY-547 and Cy5 for DY-647.
[b] The Cy3 channel was used Alexa546, staining bacterially secreted InlC, while Cy5 was used for Alexa647, staining cellular IL-8 during imaging.
[c] The two values stand for primary and secondary infection times, respectively. The same goes for pathogen dyes.

3.2 RNA Interference Protocols

Central to the siRNA screens produced by InfectX was the effort to develop unified protocols for wet-tab experiments and subsequent analysis. While this approach was successfully implemented for many aspects, some deviations among the pathogens are inevitable, while others are intentionally developed to achieve similar phenotypes. Table 3.2 summarizes some key parameters that vary between pathogens, which include seeded cell number, multiplicity of infection (MOI) and infection times, all optimized to yield infection rates that are of comparable magnitude. The target value for cell number was 1500 per well in order to create densely populated areas interspersed with some empty spaces, leading to cells living both surrounded by neighbors and on colony edges. Targeted infection rates are in the range 30–50%. Pathogen properties made it in some cases impossible to meet these goals and infection rate for *B. abortus* remained low (~5%), while being high in *B. henselae* (~90%) despite best efforts.

The usage of control wells enables diagnosis of possible problems that may occur in RNAi screens, including cytotoxicity of siRNA, low transfection yields, failure of RNAi pathway induction, dominance of non-specific responses, and therefore should be embedded in every assay plate. Three types of control experiments are typically employed: positive, negative and mock (no siRNA treatment). Positive controls are used to confirm expected response while negative controls help distinguish sequence specific from unspecific effects, and mock experiments present a baseline (Sittampalam et al. 2004).

Positive controls ideally constitute of siRNAs with known effect on the assay under investigation and are therefore often unavailable beforehand. Instead, controls to check transfection efficiency and reporter quality are usually implemented. One straightforward possibility for monitoring delivery is by targeting a gene that is vital to the cell. Kinesin family member 11 (Kif11), for example is a gene involved in cell cycle progression, the down-regulation of which is induces apoptosis. Furthermore there are mixtures of siRNAs available (e.g. AllStars Hs Cell Death Control siRNA by Qiagen) optimized for killing cells by targeting several ubiquitously expressed genes essential for cell survival. The downside of assessing transfection by killing cells is that a potentially toxic effect of the delivery mechanism itself may be masked. This can be mitigated by either performing negative control experiments (which should be done anyways) or by targeting housekeeping genes that are abundantly expressed but do not affect cell viability. Dharmacon suggests three such genes, Peptidylprolyl cis-trans isomerase B, glyceraldehyde-3-phosphate dehydrogenase and lamin A/C, for which they sell specially branded control siRNAs (as does Ambion).

Fluorescent dyes are also frequently employed in positive control experiments, typically by labeling siRNA, allowing for visual inspection of reagent localization within the cell (nuclear uptake indicates efficient transfection), or by targeting reporter genes. The latter method either allows for confirming that the reporting mechanism (usually expression of GFP or luciferase) works as intended, or establishing that siRNA transfection was successful. Again, siRNA products targeting the commonly used reporter genes are readily available from manufacturers.

Negative controls should lack homology with known targets in order to separate non-specific effects from sequence specific silencing. Such siRNAs are therefore engineered to contain a passenger strand seed region (the first 2–6 nt from the 5′ end) that matches no known gene and generally have poor overall sequence complementarity between guide strand and any known gene. One way of generating such a sequence is taking an assay siRNA and randomizing the order of nucleotides while keeping the nucleotide composition

Table 3.3: Depending on screen type and pathogen, different genes were targeted for control experiments. Abbreviations: AU, Ambion unpooled; DP, Dharmacon pooled; DU, Dharmacon unpooled; and QU, Qiagen unpooled.

	Adeno AU	Adeno DP	Adeno DU	Adeno QU	*Bartonella* AU	*Bartonella* DP	*Bartonella* DU	*Bartonella* QU	*Brucella* AU	*Brucella* DP	*Brucella* DU	*Brucella* QU	*Listeria* AU	*Listeria* DP	*Listeria* DU	*Listeria* QU	Rhino AU	Rhino DP	Rhino DU	Rhino QU	*Salmonella* AU	*Salmonella* DP	*Salmonella* DU	*Salmonella* QU	*Shigella* AU	*Shigella* DP	*Shigella* DU	*Shigella* QU	Vaccinia AU	Vaccinia DP	Vaccinia DU	Vaccinia QU
ARPC3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ATP6V1A	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Abi1				✓				✓				✓				✓				✓				✓				✓				✓
CDH4				✓				✓				✓				✓				✓				✓				✓				✓
FRAP1			✓	✓				✓				✓				✓			✓	✓				✓				✓				✓
ITGB1				✓			✓	✓			✓	✓				✓				✓				✓				✓				✓
Kill[a]				✓				✓				✓				✓				✓				✓				✓				✓
MAP3K7				✓				✓				✓				✓				✓				✓			✓	✓				✓
MET				✓				✓				✓	✓	✓	✓					✓				✓				✓				✓
MOCK	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PI4KB				✓				✓				✓				✓				✓				✓				✓				✓
PSMC3				✓				✓				✓				✓				✓				✓				✓			✓	✓
PXN			✓	✓			✓	✓			✓	✓				✓			✓	✓				✓				✓				✓
Scra 1[b]	✓				✓				✓				✓				✓				✓				✓				✓			
Scra 2[b]	✓	✓	✓	✓		✓	✓	✓		✓	✓	✓		✓	✓	✓	✓	✓	✓	✓		✓	✓	✓		✓	✓	✓		✓	✓	✓
CDC42	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓		✓	✓	✓		✓	✓	✓	✓	✓	✓	✓		✓	✓	✓		✓	✓	✓	✓
RAC1			✓	✓			✓				✓								✓	✓											✓	✓
GFP 1[c]	✓				✓				✓				✓				✓				✓				✓				✓			
GFP 2[c]	✓	✓	✓	✓		✓	✓	✓		✓	✓	✓		✓	✓	✓	✓	✓	✓	✓		✓	✓	✓		✓	✓	✓		✓	✓	✓
Kif11	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TSG101																															✓	✓
ARF1																											✓					
CBL														✓	✓																	
CFL1																							✓									
CHUK																											✓					
CLTC														✓	✓																	
DNM2														✓	✓																	
ITGAV																							✓									
NOD1																											✓					
PAK1																															✓	✓
PI4KA																											✓					
PIK3R1														✓	✓																	
PRKCA			✓	✓															✓	✓												
PSMA6																															✓	✓
Rab2								✓				✓																				
RAB7A																							✓									
SNX9			✓	✓															✓	✓												
TLN1								✓				✓																				

[a] A positive control cell killer is provided by Qiagen (AllStars Hs Cell Death Control)

[b] Scrambled siRNA sequences are provided by Ambion (Silencer Select Negative Control; Scra 1) and Dharmacon (ON-TARGETplus Non-targeting Control; Scra 2).

[c] GFP targeting siRNA sequences are provided by Ambion (Ambion Silencer eGFP; GFP 1) and Dharmacon (GFP Duplex III; GFP 2).

unchanged (i.e. scrambling the sequence). Multiple proposals are usually generated and subsequently checked for applicability by sequence alignment to the target genome. While scrambling has the advantage that a possible effect of nucleotide composition is removed, it is infeasible for large-scale screens and often a set of predefined sequences sold by manufacturers (for example Silencer Select Negative Control from Ambion, ON-TARGETplus Non-targeting Control siRNAs from Dharamcon or AllStars Negative Control siRNA from Qiagen) is used instead (while still being called scrambled controls).

Table 3.3 lists the siRNA control experiments that were performed throughout all screens and indicates the set of controls each screen contains. Common to all pathogens and all libraries are the previously mentioned positive control Kif11, one of two GFP targeting sequences, scrambled siRNAs, as well as mock experiments. Additionally, several controls for general infection mechanisms were included in most screens, including ATP6V1A (a H^+ transporting ATPase, responsible for acidification of endocytic and lysosomal vesicles), as well as ARPC3 (Arp2/3), and Cdc42, both of which are part of actin-dependent processes surrounding pathogen uptake and ABM. Some controls however are specific to a subgroup of pathogens or single pathogens and include the following (grouped by pathogen).

***Bartonella/Brucella*:** The small GTPase Rab2 is required for anterograde ER–Golgi transport, capable of interacting with RicA of *B. abortus* (Barsy et al. 2011) and TLN1 (talin-1) has been determined to be necessary for invasome formation in *B. henselae* infection via β_1 signaling (Truttmann et al. 2011).

***Listeria*:** During endocytosis of *L. monocytogenes*, the ubiqutin ligase Cbl is recruited to the site of entry and seems to be involved in InlB mediated, clathrin dependent (CLTC encodes the clathrin heavy chain 1 protein) bacterial uptake (Veiga and Cossart 2005). DNM2 (dynamin-2) is a further protein involved in host entry and PIK3R1 is a phosphatidylinositol 3-kinase, downstream to Met, the cellular receptor to InlB.

***Salmonella*:** The actin-modulating protein CFL1 (cofilin-1), responsible for actin depolymerization, the cellular integrin receptor ITGAV and Rab7A, a small GTPase that regulates vesicular trafficking, have all been shown to be hits in a salmonella invasion screen (Misselwitz et al. 2011).

***Shigella*:** Down regulation of ARF1 (a guanosine triphosphate binding protein involved in vesicle trafficking) and phosphatidylinositol 4-kinase PI4KA has been suspected of interfering with pathogen entry, while suppression of CHUK/NOD1 (both involved in NF-κB signaling) may inhibit IL-8 production by uninfected bystander cells, thereby possibly promoting infection (Kasper 2012).

Adenovirus/rhinovirus: SNX9 regulates dynamin assembly and is therefore crucial to viral endocytosis and PRKCA is a serine-/threonine-specific protein kinase responsible for a wide array of regulatory signals. PRKCA might be involved in influenza virion buddying and has been implicated in playing a role in intracellular proliferation of hepatitis (Kanehisa and Goto 2000).

Vaccinia virus: The serine-/threonine-specific protein kinase PAK1 is targeted by Rac1/Cdc42 and has been shown to be required for MV entry (Mercer and Helenius 2008). Up-regulation of PSMA6 (a proteasome complex component) enables evasion of intracellular immune surveillance (Zhou et al. 2014) and TSG101, ubiquitin-conjugating enzyme, hampers EV production (Honeychurch et al. 2007).

Many of these pathogen specific positive control targets are not well established and while some have previously been identified and validated, others represent best guesses and might not serve their purpose particularly well. Controls wells are typically located at the plate border (rows A and P; columns 1, 2, 23 and 24; although other control layouts also exist) and the different control experiments are replicated multiple times on each plate.

For all screens, siRNA transfection was carried out by adding 25 µl of RNAiMAX/DMEM (0.1 µl/24.9 µl) transfection agent to 1.6 pmol siRNA diluted in 5 µl RNase-free ddH_2O contained in each of the 384 wells per screening plate. After 1 h incubation at room temperature, the required number of cells were added (see table 3.2), suspended in 50 µl DMEM/16% FBS. Plates were subsequently incubated for 72 h at 37 °C and 5% CO_2, followed by the pathogen specific infection procedure (see section A.1 for details). Following infection, cells were fixed in paraformaldehyde (PFA) and stained for DNA, F-actin and additional pathogen specific markers (see table 3.2). Plates were sealed prior to imaging.

3.3 Image Acquisition and Data Processing

Imaging was performed both at the University of Basel and the Light Microscopy and Screening Center of ETH Zürich, using ImageXpress micro (IXM) HCS wide-field microscopes from Molecular Devices, equipped with Thermo Scientific CataLyst Express robotic plate handlers capable of storing and serving up to 45 plates. Lumencor Spectra X solid-state light engines (LED light sources), 10x S Fluor objective lenses by Nikon with a numerical aperture of 0.45 and Photometrics CoolSNAP HQ 14-bit CCD cameras, resolving 1392 × 1040 pixels (individual pixel size of 6.45 µm × 6.45 µm), complete the hardware setup. Channel selection is assay specific and stain dependent Semrock

filters (DAPI/Hoechst, GFP/FITC/Alexa488, Cy3, Cy5, Quadband DAPI-GFP-mCherry-Cy5) are employed (see table 3.2 for details).

Wells are divided into 3×3 grids for most plates while some 2×3 site images exist too, with no spacing and no overlap, and Molecular Devices MetaXpress High-Content Image Acquisition and Analysis Software was used for recording images. Software parameters include: no gain, well bottom as autofocus target, site-specific autofocusing, enabling of laser-based focusing and no shading correction. For each imaging channel, focus Z-offset was selected manually and exposure time was automatically calculated. In cases of poor dynamic range or overexposure, manual correction to exposure time was applied. Upon imaging, data was transferred to iBrain2/screeningBee (Rouilly et al. 2012) for further processing.

3.3.1 Data Handling (iBrain2/screeningBee)

In case of microscopy based siRNA screening experiments, a complex task of data handling and processing follows the imaging stage. A wealth of data is generated by imaging devices[3] which has to be accessibly and redundantly stored. Furthermore, analysis of image data is a processing intensive task that quickly becomes reliant on high performance computing (HPC) resources, entailing specialized requirements due to the oftentimes shared and centralized nature of such systems. The authors of iBrain2 summarize the key steps in RNAi high content screening data processing as follows (Rouilly et al. 2012):

1. **Data acquisition.** The raw data of siRNA screens is produced as digital images by microscopy. Acqusition times of several hours per plate are typical and a single plate yields ~20 GB of data.
2. **Permanent storage.** Due to infeasibility of re-screening plates, all raw data has to be stored in a sufficiently redundant manner. Furthermore, the permanent data store has to be able to serve portions of the dataset quickly and efficiently.
3. **Temporary data staging to HPC.** In order for the compute cluster not to be bound by network latency, it is often necessary to stage the data to be analyzed to local scratch space. This step becomes superfluous whenever the permanent storage system is directly integrated in the cluster's high-speed network.

[3] A genome-wide screen (~27000 individual experiments) in 384 well format involves 70–100 plates depending on the number of controls per plate. Each plate yields 10000–14000 images (384 wells, 9 sites and 3–4 channels), leading to 700000–1400000 individual images and requiring multiple terabytes of storage.

4. **Data analysis.** Many aspects of processing a large number of images are embarrassingly parallel and a cluster environment is ideally suited to tackle this computation intensive task.

5. **Permanent storage of results.** Some results produced by the analysis procedure will be saved back to the permanent storage system. While it may be sensible to save storage space and carry out some procedures on the fly, this is not feasible for all analysis routines.

6. **Publication and archiving.** Upon completion of the project, some data will be made availably publicly and all data worth keeping is moved to an archival system capable of cheap long-term storage where quick retrieval is unimportant.

The requirements outlined above led to the development of iBrain2 as a modular workflow manager capable of setting up reproducible procedures. The software is implemented in Java and uses an XML-based format for defining workflows. Furthermore, it interfaces with other open source projects, such as openBIS (Bauch et al. 2011) which can be used as storage system and CellProfiler (Carpenter et al. 2006; Kamentsky et al. 2011), a popular tool for analyzing cell based image material. Within InfectX, not iBrain2 itself is used, but a derivative thereof called screeningBee, featuring tighter openBIS integration and a large set of CellProfiler extensions.

3.3.2 Image Analysis (CellProfiler)

Prior to the development and release of CellProfiler in 2006, large-scale screens were routinely evaluated though tedious and labor intensive visual inspection by expert biologists. While humans are able to perceive and contextualize image data in ways that computers still struggle to emulate, a lot of detail is lost though human interpretation. Small differences are hard to spot, leading to subtle patterns being missed and humans focus on a handful of features, not being able to take into account the plethora of information encoded in crowded images. Furthermore, human study of image data leads to qualitative evaluations, while computational analysis yields a potentially large number of quantitative measures. Lastly and perhaps most importantly, human based assessment of imagery coming from a project the size of InfectX is simply infeasible. In the words of Carpenter et al., the authors of CellProfiler, consequences of implementing computational image analysis for HTS are as follows:

> With the successful application of sophisticated image analysis methods, the bottleneck of image-based genome-wide screens is now moving downstream to data visualization, exploration, and statisti-

> cal analysis in order to accommodate the number and richness of measurements that result from image-based genome-wide assays.

CellProfiler is an open-source software solution catering to the needs of HTS screening initiatives and providing a customizable, modular platform for cellular image analysis. CellProfiler processing is based on modules which are placed in sequential order to form a pipeline, usually starting out with image processing, followed by object identification and concluded by calculation of feature measurements on these objects. The pipeline's modules and settings can be saved to a configuration file, in order to ensure reproducibility and transferability of workflows. The original version of CellProfiler was written in MATLAB and a more recent version 2 has been ported to python. InfectX procedures are based on version 1 (Carpenter et al. 2006).

The InfectX image analysis routine starts out with metadata collection and image quality assessment modules, followed by shading correction. Detecting out of focus images, wells that show signs of experimental error or problematic plates, however is still largely carried out by humans. Position dependent illumination and sensor inhomogeneities need to be corrected for, as they lead to decreased sensitivity in darker areas and make comparison of intensity based features between objects belonging to different areas of the image problematic.

Shading correction as offered by a default module of CellProfiler, was determined to consistently underestimate light fall-off, prompting the development of an improved algorithm for InfectX. Instead of using all available intensity information to detect uneven illumination, images are separated into fore-/background and only foreground intensities are considered for shading correction. While this method yields better estimates, corrections are still conservative and currently, fore-/background separation is performed manually (once per plate/channel). A fully automated procedure has been developed by Smith et al. (2015) and again improves on the results of separated images. Re-calculation of shading correction is currently being carried out but is not yet available for most of the kinome data this project is focused on.

Illumination corrected images are stitched together, channel separated and forwarded to CellProfiler modules for segmentation and feature calculation. Cell identification is a hard problem, especially if clumping occurs. Whenever objects are present that can easily be identified, such as nuclei (providing DNA was stained), these are identified first (primary objects), facilitating identification of secondary objects, e.g. cells. In case of clumped cells, a three step algorithm is employed: first, clumps are recognized by segmentation, then dividing lines are sought and defined, followed by the final step, attempting to correct for erroneous or missing splits by removing or merging some resulting objects, based on measurements such as size or shape.

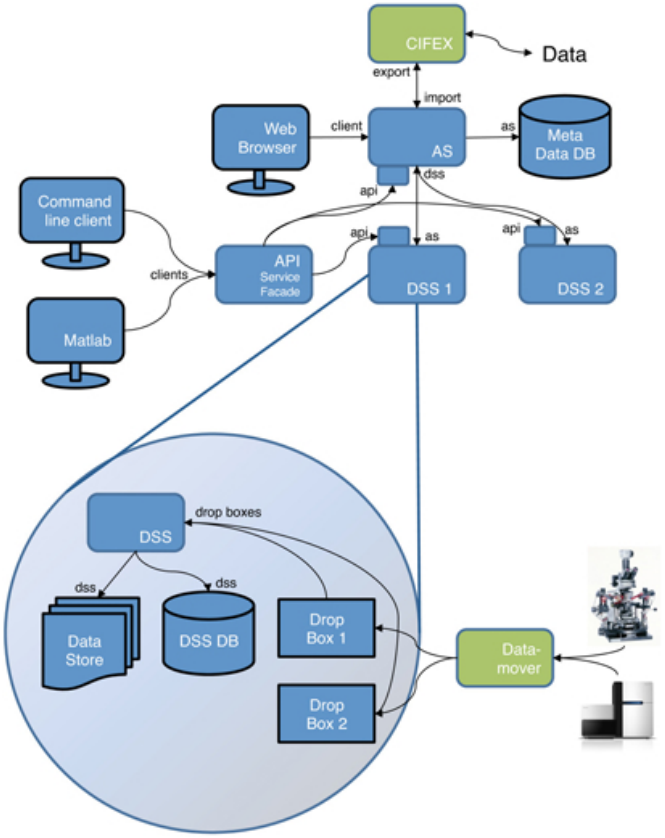


Figure 3.2: An openBIS instance is deployed as a service consisting of an application server (AS) and one or more data store servers (DSSs). At both levels, clients can interact with the system to query, fetch or deposit data. Furthermore a browser based graphical user interface is available alongside an extensive set of APIs in order to integrate the system into many possible environments (Bauch et al. 2011).

Several modules developed by InfectX are used during the feature extraction stage, extending CellProfiler functionality to specific requirements including invasome and bacterial aggregate detection, more efficient neighbor feature measurement and collection of various properties per sub-cellular object. The features calculated by the CellProfiler pipeline are explored more in-depth in section 3.4. All results produced by image analysis are saved to an openBIS instance.

3.3.3 User Accessible Data Storage (OpenBIS)

The authors of openBIS (open Biology Information System) argue that the availability of a domain specific information management system plays an important role in data-driven biological research, serving as a basis for management of large amounts of experimental data and acting as both a source and a sink for analysis procedures. Furthermore, such a system should provide a user interface capable of visualizing complex dependency structures and a set of APIs for accessing the stored data programmatically. Finally, it should be possible to make a subset of the data available to the public, providing a simplified interface of the browsing and searching capabilities (Bauch et al. 2011).

In order for openBIS to be applicable to the broad range of scenarios it is intended for, extensibility and transparent interfaces are paramount, as is scalability and performance. To address the latter aspect, separation of bulk data from metadata is a guiding principle. Both ingestion of high-throughput, high-content data as well as serving it to users and analysis routines is CPU, I/O and network intensive and therefore should not be performed on the same system as user-facing metadata queries which only deals with comparably inexpensive tasks that scale well. Additionally, scalability requires that bulk data storage may be distributed among several systems.

With these requirements in mind, openBIS is designed to consist of an application server (AS), dealing with metadata, serving user queries such as searches and visualizing dependencies via the browser based graphical user interface, as well as one or several data store servers (DSSs) that handle the bulk data. Communication between the two layers is asynchronous (see figure 3.2).

All datasets in openBIS are immutable. While this is potentially wasteful in terms is storage resources, it can be argued that storage space is cheap and such an architecture improves traceability of constant evolution of analysis procedures and corresponding results. Datasets are subject to a hierarchical structure paradigm consisting of the entities *(1) Data Space*, *(2) Project*, *(3) Experiment*, and *(4) Sample*. In the example of siRNA HTS data, a sample represents a plate and associated datasets include raw images, feature measurements and infection scores. Metadata can be provided either in unstructured form which is handled as attachments associated with project, experiment or sample entities, or as structured objects (consisting of name, label, description and value) that are linked to experiments, samples and datasets.

Data storage by a DSS follows a hybrid model consisting of a relational database, storing index information, file metadata and selected results, in addition to flat-file data store which can be distributed between several file systems. Management of fill state of the associated storage shares and data distribution is

all governed by the individual DSSs. Some data presentation tasks handled by DSSs include data visualizations such as plate heatmaps and assembly of multichannel images, as well as overlay of analysis results (e.g. segmentation).

Ingestion of new data can either be triggered by web import, custom software talking to the corresponding API or simply by adding to folders that are monitored by a DSS as dropboxes. Responding to write activity to these observed locations, the highly customizable ETL (extract, transform, load) routine extracts metadata, creates datasets annotated with metadata in the AS database, links them to the appropriate entities and adds the new datasets to the data store. Export of data is either browser based or can be handled by provided command line tools, accessing openBIS via a Java service facade. Two separately designed tools are integrated for usage with openBIS: CIFEX and Datamover. The former enables browser based transfer of large files and supports interruption of transfers as well as checksumming for file integrity, while the latter can be used to automatically transfer data off of an instrument computer into a dropbox folder via SSH or rsync protocols.

3.4 Single Cell Feature Data

The set of accessible single cell features varies among screens and pathogens to some extent. While there are certain general features, such as geometry, intensity and texture of nuclear and cellular objects available for every plate dataset, others, including invasome and IL-8 features are specific to a subset of the pathogens, while others still, such as certain neighbor features or Vononoi cell segmentation were only recently added to the feature extraction pipeline and therefore have not yet been calculated for all screens. Such discrepancies are not treated by this overview which focuses on general aspects of currently available single cell features.

Nomenclature of single cell features follows a hierarchical pattern consisting of an object specifier, a measurement group, the measurement name itself and the imaging channel it was applied to, as follows:

```
<object>.<group>_<measurement>_<channel>
Cells.Intensity_MedianUpperTenPercentIntensity_CorrDNA
```

In the above example, the measured objects are cells, the feature is of intensity type, the feature name indicates that it represents the mean of intensities above the upper decile and the channel description reveals that DNA stain values are used. Throughout the different screens, 14 single cell feature objects are available:

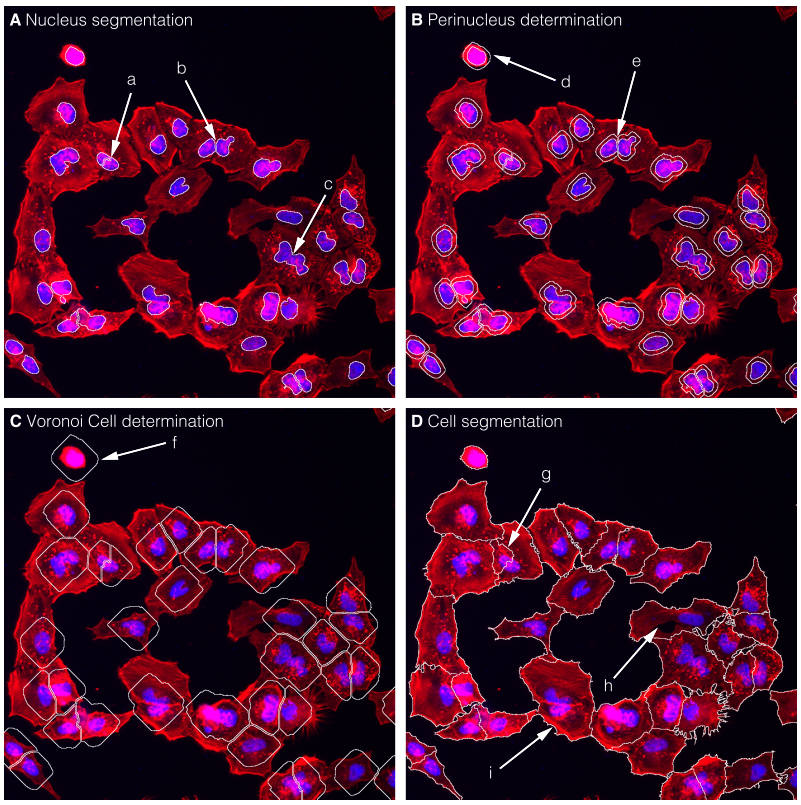


Figure 3.3: Object identification is started with easily recognizable nuclei, which become primary objects and serve as basis for detecting secondary objects. Problems with nuclei detection include superfluous splitting (a) and failure to separate some touching objects (c). Mostly results are acceptable and some touching nuclei are correctly separated (b). Perinuclei extend the nuclear border and therefore suffer from problems introduced in nuclear recognition. Further issues include overlap with extracellular space (d) and small perinuclear area in densely populated regions (e). Voronoi cells further extend the nuclear area and often grow beyond cell boundaries (f). Finally, cell segmentation is a hard problem due to irregular shapes, variability in size and inhomogeneities in actin distribution (h). Results suffer from falsely identified nuclei (g and i). Images are reproduced from InfectX *Salmonella* screening data.

Nuclei: Using the DNA channel, nucleus recognition is fairly accurate owing to generally uniform morphology of nuclei, uniform distribution of DNA and good contrast of DNA staining. The current implementation for segmentation is based on Otsu's method which converts a monochromatic image into a binary image by finding an appropriate threshold, effectively separating foreground from background. Objects that have an Euler characteristic $\chi \leq 0$ are filled and apparently clumped objects are split. Finally, objects outside of an allowed size range are discarded (mostly dust particles which are too big, as well as pathogen DNA which yields objects that are too small). Nuclei represent primary objects with serve as basis for the detection of secondary objects (see figure 3.3, A).

PeriNuclei: The perinuclear region is defined as an extension of the nuclear border by 8 pixels and removal of the nucleus itself. As some pathogens establish their replicatory niche in this region it lends itself to closer investigation. Furthermore the border is only dependent on the reliably established nuclear object. Potential pitfalls include extension of the perinuclear region into extracellular space or neighboring cell when the nucleus is located close to the plasma membrane, as well as reduced area whenever cells are clustered, as perinuclear regions may not overlap (see figure 3.3, B).

VoronoiCells: Simultaneously extending all nuclei by maximally 24 pixels or until a neighboring border is met leads to Voronoi cells. This type of defining cell bodies is very prone to extending into extracellular space, as the actin channel is not taken into account (see figure 3.3, C).

Cells: The cell body border is determined by propagation segmentation based on thresholding whenever corresponding staining is available. Starting from the nucleus, the cytoplasmic region is extended until the intensity gradient changes abruptly, indicating either the beginning of background, or the boundary to a neighboring cell. Non-uniform actin intensity patterns, irregular shape, large variability in size and frequent touching of neighbors make this a hard problem and results can be unreliable. Common to all secondary objects, errors in determining primary objects will propagate and lead to cells erroneously being cut up or aggregated. Furthermore, borders to neighbors are frequently not determined correctly, but significant discrepancies among human expert labeling are to be expected in such situations as well (see figure 3.3, D).

ExpandedNuclei: Serving as precursors to PeriNuclei objects, these objects are not further quantified and only have coordinates and number of perinuclear children (almost exclusively 1) associated.

Bacteria/Viruses/Pathogen: Due to the heterogeneity between pathogens, detection has to be assay specific. Generally, pathogens are primary objects,

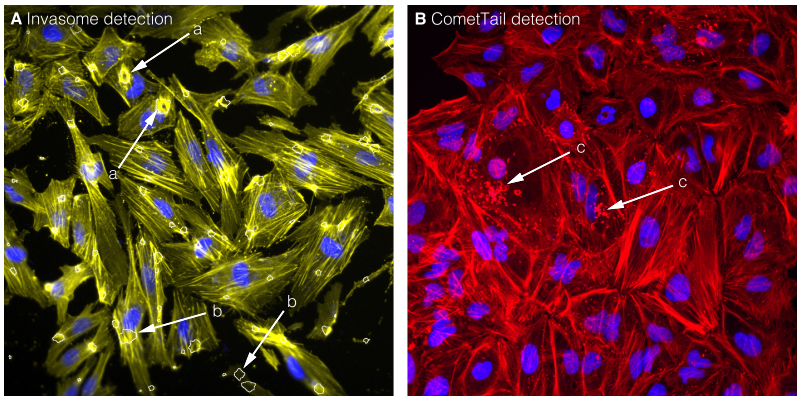


Figure 3.4: Both CometTails and Invasomes are detected on actin channels, which poses difficulties due to noisy background. Invasomes are characterized by their ring-shaped morphology (a), while CometTails appear as actin streaks (c). Currently invasome detection yields many false-positives (b), which is problematic due to their inclusion in infection scoring (see figure 3.7). Images are taken from InfectX *Bartonella* (A) and *Listeria* (B) screens.

meaning they are and not derived from cell nuclei. In *Bartonella* and *Brucella* screens, large and small pathogen clusters are segmented using Otsu's method on the pathogen channel. Individual bacteria in *Listeria* screens, SCVs in *Salmonella* screens and small/medium pathogen clusters in both *Shigella* and rhinovirus screens are identified using wavelet decomposition of the pathogen channel, while for adenovirus screens, detection of large pathogen aggregates relies on propagation outwards from nuclei, much like cell body identification but using the pathogen channel. For vaccinia virus, direct pathogen segmentation is currently not possible.

IntBacteria/ExtBacteria: In *Bartonella* assays, different stains were used for intracellular (GFP) and extracellular (Cy5) bacteria which consequently can be distinguished during image analysis. This is helpful to prevent bacteria located on top or underneath a cell from being considered intracellularly located.

BlobBacteria: Specific to *Brucella* screens, these large bacterial aggregates do not have any real measurements associated, except coordinates.

Invasomes: As one of two internalization structures in *Bartonella* infection, invasomes are characterized as actin surrounded, large bacterial aggregates. Exploiting this morphology, the current invasome detection uses a ring template to search the actin channel for candidates and exploits the intensity difference to the cell mean for confirmation. Unfortunately, this yields

many false positives. Improvements are being discussed but have yet to be implemented (see figure 3.4, A).

IL8: In *Shigella* assays, cellular IL-8 was stained in order to study proinflammatory response of uninfected bystander cells. Due to these objects having their own channel, segmentation is fairly straightforward.

CometTails: Specific to *Listeria* screens, the telltale signs of ABM can be identified as dense strokes on the actin channel. CometTails contain only few measurements beyond their location (see figure 3.4, B).

Neighbors: While not representing detected objects themselves, features involving identities of adjacent objects are summarized within this category. Features of this category allow construction of both a binary adjacency matrix and a weighted adjacency matrix corresponding to each object type, with weights representing the length of common border.

Once images are segmented into objects, measurements are applied to the individual image areas. Dependent on object class, the number and types of available features differ, sometimes owing to systematic causes and at other times, only due to differences in implementations of per-object feature recognition. The most commonly available features are grouped into categories *AreaShape*, *Intensity*, *Texture*, *RadialDistribution*, *Location*, *Neighbors/IdentityOfNeighbors/PercentTouchingNeighbors* and *Parent/Children*.

Features belonging to the *AreaShape* group are not applied to a specific imaging channel but serve to summarize geometric properties of detected objects (*cf.* table 3.4). When applied to actin channel objects that cannot reliably be segmented, such as cell bodies, some caution has to be exerted, as these features are strongly affected by upstream errors. Nevertheless, this group constitutes features that are among the most frequently used for analysis procedures. Certain infection phenotypes, for example membrane ruffling encountered during trigger-type host entry mechanisms are captured by some of these features (*FormFactor* is able to measure raggedness).

Intensity features mostly consist of summary statistics being applied to all or a portion of per channel intensity data within a given object. The functions used to describe the underlying distributions are sum (Integrated), lower quartile, maximum (*Max*), mean, median, minimum (*Min*), standard deviation (*Std*), upper quartile, upper vigintile (*UpperFivePercent*), upper decile (*UpperTenPercent*) and upper 50-quantile (*UpperTwoPercent*). Additional features are generated by using certain functions only on the object border pixels, yielding IntensityEdge features (e.g. *IntegratedIntensityEdge*) and by only including the upper 10%, 5% and 2% of the data. Weighted features are calculated by extracting weights from a corresponding grayscale image and multiplying individual in-

Table 3.4: List of AreaShape features with corresponding descriptions. Some information is taken from the CellProfiler manual (Carpenter et al. 2006).

Feature name	Feature description
Area	The number of pixels enclosed by the object border.
Eccentricity[a]	Ratio of the distance between the foci and major axis length of the corresponding ellipse. The value is between 0 (circle) and 1 (line segment).
EulerNumber	1 minus the number of holes within the object, assuming 8-connectivity.[b]
Extent	Area of the object divided by the area of the bounding box.
FormFactor	Calculated as $4\pi \cdot area/perimeter^2$. Equals 1 for a perfectly circular object.
<Spec>AxisLength[a]	Spec = {Major, Minor}; The major/minor axis length (in pixels) of the corresponding ellipse.
Orientation[a]	The angle (ranging from $-90°$ to $90°$) between the x-axis and the major axis of the corresponding ellipse
Perimeter	The total number of pixels around the object boundary.
Solidity	Proportion of the pixels in the convex hull that are also part of the object, i.e. $area_{object}/area_{conv.hull}$.
<Sub[c]>Area	The total number of pixels enclosed by all subcellular pathogen object borders.
<Sub[c]><Stat[d]>Distance-ToNuclNorm	Distances between the nucleus and all subcellular pathogen objects are normalized and a summary statistic is applied.
<Sub[c]><Stat[d]>Distance-ToNucl	Distances from nucleus to all subcellular pathogen objects are calculated and a summary statistic is applied.
<Sub[c]><Stat[d]>Pairwise-Distance	A summary statistic is applied to all pairwise distances between subcellular pathogen objects.
<Sub[c]><Stat[d]>Shortest-Distance	For each subcellular pathogen object, the shortest distance among all distances to other subcellular objects is selected and a summary statistic is applied to the collection of shortest distances.

[a] The corresponding ellipse is defined as the best fitting ellipse in the sense that it has identical second moments compared to the original object (Rocha, Velho, and Carvalho 2002).

[b] The neighborhood of a central pixel comprises of all 8 surrounding pixels, touching either a corner or an edge of the central cell.

[c] Naming of these features currently is fairly heterogeneous with possible prefixes *PerObj*, *SubCellBacteria*, *SubCellViruses*, *SubCell*. All are based on the set of subcellularly located pathogens.

[d] The statistics applied in order to describe the underlying distributions are mean, median, lower and upper quartiles, standard deviation and sum.

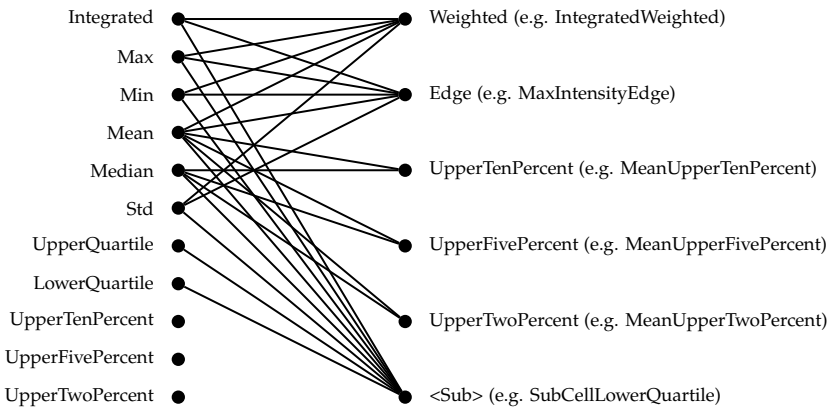


Figure 3.5: Common summary statistics that are applied to intensity data, along with their possible modifiers. The subcellular intensity features have different prefixes, including SubCellBacteria, SubCellCometTails, SubCellViruses, PerObject and SubCell.

tensity values before applying the summary statistic. Restricting the data to subcellular pathogen objects yields features such as *SubCellLowerQuartile*. For a complete list of modifiers and implemented combinations, refer to figure 3.5. The only currently available intensity feature that is not shown in 3.5 is *MassDisplacement*, which quantifies the distance between the centers of gravity in the gray-level representation of the object and the binary representation of the object.

All texture measurements are either based on Haralick features (Haralick, Shanmugam, and Dinstein 1973) or Gabor wavelet features (Gábor 1946), as implemented in CellProfiler and are calculated per image channel. The gray-level co-occurrence matrix G_k with dimensions $n \times n$ where n is the number of gray levels serves as basis for Haralick's texture features (Carpenter et al. 2006).

$$G_k = \begin{bmatrix} p(1,1) & p(1,2) & \cdots & p(1,n) \\ p(2,1) & p(2,2) & \cdots & p(2,n) \\ \vdots & \vdots & \ddots & \vdots \\ p(n,1) & p(n,2) & \cdots & p(n,n) \end{bmatrix} \tag{3.4}$$

Element $p(i,j)$ is calculated by counting the number of times, a pixel with value i is adjacent to a pixel with value j, divided by the total number of com-

parisons. Consequently, $p(i,j)$ can be thought of as representing the probability of two gray level values appearing next to each other. Moreover, by defining the neighborhood of a pixel as all 8 surrounding units touching either a corner or an edge of the central cell (8-connectivity), 4 different types of adjacency are possible (horizontal, vertical, top left to bottom right diagonal and top right to bottom left antidiagonal), yielding 4 co-occurrence matrices $G_k, k \in \{1,2,3,4\}$. First, some notation to more compactly represent statistics developed by Haralick, Shanmugam, and Dinstein:

$$p_x(i) = \sum_{j=1}^{n} p(i,j), \qquad p_y(j) = \sum_{i=1}^{n} p(i,j),$$

$$p_{x+y}(k) = \sum_{\substack{i=1 \\ i+j=k}}^{n} \sum_{j=1}^{n} p(i,j), \qquad p_{x-y}(k) = \sum_{\substack{i=1 \\ |i-j|=k}}^{n} \sum_{j=1}^{n} p(i,j),$$

$$HX = -\sum_{i=1}^{n} p_x(i) \log\left(p_x(i)\right), \qquad HY = -\sum_{i=1}^{n} p_y(i) \log\left(p_y(i)\right),$$

$$HXY1 = -\sum_{i=1}^{n} \sum_{j=1}^{n} p(i,j) \log\left(p_x(i) p_y(j)\right),$$

$$HXY2 = -\sum_{i=1}^{n} \sum_{j=1}^{n} p_x(i) p_y(j) \log\left(p_x(i) p_y(j)\right)$$

Means and standard deviations of p_x and p_y are represented by μ_x, μ_y, σ_x and σ_y, respectively, while HX and HY are the entropies of p_x and p_y. Of the 14 features developed by Haralick, Shanmugam, and Dinstein, 13 are included with CellProfiler:

$$\text{AngularSecondMoment} \qquad f_1 = \sum_{i=1}^{n} \sum_{j=1}^{n} p(i,j)^2 \tag{3.5}$$

$$\text{Contrast} \qquad f_2 = \sum_{k=0}^{n-1} k^2 \left(\sum_{\substack{i=1 \\ |i-j|=k}}^{n} \sum_{j=1}^{n} p(i,j) \right) \tag{3.6}$$

$$\text{Correlation} \qquad f_3 = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} (ij) p(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y} \tag{3.7}$$

$$\text{Variance} \qquad f_4 = \sum_{i=1}^{n}\sum_{j=1}^{n}(i-\mu)^2 p(i,j) \tag{3.8}$$

$$\text{InverseDifferenceMoment} \qquad f_5 = \sum_{i=1}^{n}\sum_{j=1}^{n}\frac{1}{1+(i-j)^2}p(i,j) \tag{3.9}$$

$$\text{SumAverage} \qquad f_6 = \sum_{i=2}^{2n} i p_{x+y}(i) \tag{3.10}$$

$$\text{SumVariance} \qquad f_7 = \sum_{i=2}^{2n}(i-f_8)^2 p_{x+y}(i) \tag{3.11}$$

$$\text{SumEntropy} \qquad f_8 = -\sum_{i=2}^{2n} p_{x+y}(i)\log\left(p_{x+y}(i)\right) \tag{3.12}$$

$$\text{Entropy} \qquad f_9 = -\sum_{i=1}^{n}\sum_{j=1}^{n} p(i,j)\log\left(p(i,j)\right) \tag{3.13}$$

$$\text{DifferenceVariance} \qquad f_{10} = \sum_{i=0}^{n-1} i^2 p_{x-y}(i) \tag{3.14}$$

$$\text{DifferenceEntropy} \qquad f_{11} = -\sum_{i=0}^{n-1} p_{x-y}(i)\log\left(p_{x-y}(i)\right) \tag{3.15}$$

$$\text{InfoMeas1} \qquad f_{12} = \frac{f_9 - HXY1}{\max\{HX, HY\}} \tag{3.16}$$

$$\text{InfoMeas2} \qquad f_{13} = \left(1-\exp\left[-2(HXY2-f_9)\right]\right)^{\frac{1}{2}} \tag{3.17}$$

The individual statistics are averaged over the four co-occurrence matrices to yield a single value per object per imaging channel. Computation of Haralick features is comparably inexpensive and the resulting texture characterization has successfully been used in image-based classification scenarios. Gabor filters on the other hand are more complicated and their application can incur signinificant computational cost. They quantify striped texture that is parallel to a given angle and sets of Gabor filters at certain angles are frequently used for pattern recognition tasks such as optical character recognition or fingerprint recognition. The directions for which Gabor filters are applied to segmented objects are the x- and y-axes, yielding the features *GaborX* and *GaborY*.

RadialDistribution features contain 3 measurements that are applied to each image channel. Given an object, it's area is divided into bins formed by concentric rings, where the center is the point that is the farthest from any edge. FracAtD records the fraction of total stain contained in a given bin, *MeanFrac* normalizes the fraction of total stain with the fraction of pixels of a given bin and *RadialCV*

determines the coefficient of variation of intensity within a ring, calculated over 8 slices. Features of this category are calculated for invasome objects (4 bins) in *Bartonella* screens and are used for infection scoring. Invasomes are characterized by a ring like appearance and the concentration of intensity within this ring can be detected by the *FracAtD* feature.

Coordinates are stored as location features by calculating the center of mass for each object (*Center_X*, *Center_Y*), using the top left image corner as origin, while *Parent/Children* features represent hierarchical relationships between objects (e.g. nucleus to cell). All child features are composed as *<Parent>.Children_<Child>_Count* and summarize the number of objects of a given child type that are contained within objects of a given parent type. When a child object overlaps with multiple parent objects, it is counted multiple times. Examples include the number of invasomes within a cell or the number of virions within a nucleus.

Parent features are named as *<Child>.Parent_<Parent>* and *<Child>.Parent_<Parent>_OverlapPercent*. The former type holds indices of all parent objects which is only a single number in case of child objects that are completely contained within their parents, but may also be a vector of numbers whenever overlap with multiple parent objects occurs, while the latter feature type quantifies the extent of overlap as a percentage. Vector valued Parent features were only recently added to the analysis pipeline and are therefore currently not available in all screens. The previous procedure only considered the largest child and therefore yielded a scalar value per object.

Features belonging to the Neighbors category are *NumberOfNeighbors*, *PercentTouching*, *FirstClosestObjectNumber*, *FirstClosestXVector*, *FirstClosestYVector*, *SecondClosestObjectNumber*, *SecondClosestXVector*, *SecondClosestYVector* and *AngleBetweenNeighbors*. In order to correct for slight segmentation errors, objects are expanded by a fixed number of pixels (2 for objects such as *Cells* and *Bacteria*, 8 for *VoronoiCells*) prior to being analyzed for their neighborhood. *NumberOfNeighbors* counts the total number of neighboring object of the same type, *{First, Second}ClosestObjectNumber* store indices of the respective objects, while *PercentTouching* reports the length of common border with all neighboring objects after object expansion has been performed. Distances are available in the form of *{First, Second}Closest{X, Y}Vector* and *AngleBetweenNeighbors* holds the angle formed by connection the center of the current object with its closest and second closest neighbors.

Two final groups of features are named *IdentityOfNeighbors* and *PercentTouchingNeighbors*. These are different from most previous measurement types (the only exception being parent–child relationships with multiple children) in that they are vector-valued per object. The same object expansion rules as in Neigh-

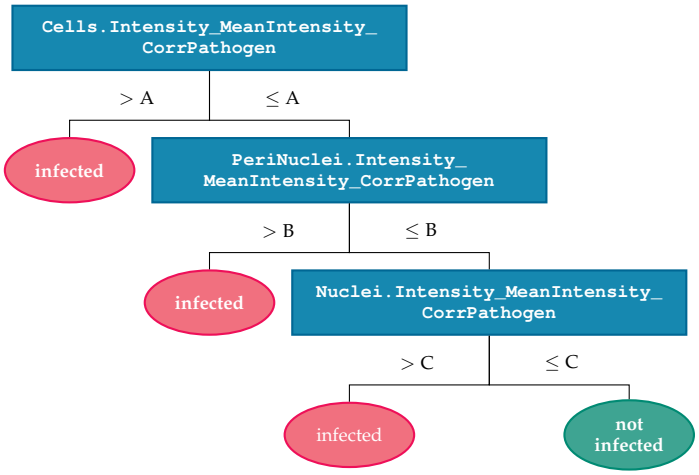
bors category features apply and *IdentityOfNeighbors* stores the respective indices while *PercentTouchingNeighbors* holds the length of common border per neighbor. The features of this group can be used to generate binary and weighted adjacency matrices, with weights representing the extent of inter-object interface.

3.5 Infection Scoring

Reliable identification of infection is central to image-based siRNA screens involving pathogens, as the per well infection index (number of infected cells divided by total number of cells in the given well) is the phenotype of main interest. Binary predictors were developed both in the form of support vector machines (SVMs) and decision trees. Currently the most reliable results are achieved using decision tree infection scoring (DTIS) but for quality control purposes it is important to routinely compare results obtained with different methods in order to spot possible problems as indicated by discrepancies.

Classification by an SVM yields the $(p-1)$-dimensional hyperplane from p-dimensional data points that best separates the data into two groups in the sense that the margin between data and plane is maximized. Sample points that lie on the margin are called support vectors. As a supervised learning procedure, training data has to be available (e.g. obtained by expert labeling) in order to produce a model that can subsequently be used to predict the category of new data instances. CellClassifier, the software used for SVM classification is described in Rämö et al. (2009). For pathogens that exhibit a clear binary infection pattern (for example *S.* typhimurium or vaccinia virus), good result can be obtained using this method (>99% accuracy), but whenever the infection phenotype is more gradual (for example *L. monocytogenes*), result are not always satisfactory. For all pathogens, 3–5 features were hand-picked and plate-wise Z-scored prior to SVM learning.

Decision trees are a classification method that can be visualized by a binary tree with internal nodes serving as decisions and external nodes representing outcomes. Logically, this corresponds to a set of AND/OR-linked statements and geometrically, the decision boundary is no longer linear (as in the case of SVMs), but piecewise linear with segments being parallel to the coordinate axes. Applying this scheme to infection classification necessitates defining a set of features and finding suitable thresholds. Once a good model has been identified, application and interpretation are straightforward and results have proven to be reliable. One down-side of DTIS is that decision thresholds are affected by plate-specific parameters (e.g. quality of staining, microscope illumination) and therefore have to be adjusted on plate-by-plate basis. The parameter sets



A	0.115	0.105	0.155
B	0.085	0.080	0.130
C	0.075	0.070	0.120
Abundance	76%	19%	5%

Figure 3.6: For adenovirus infection scoring, the decision tree classifier checks if enough pathogen is detected within the cell body, the perinuclear region or the nucleus. The threshold decreases as the region of interest concentrates on areas associated which are progressively involved in infection. Due to experimental sources of noise, the set of boundaries has to be determined plate-wise. In case of adenovirus experiments, the three parameter sets shown suffice to cover all currently analyzed plates while yielding satisfactory classification. The last line indicates the coverage of each set of parameters.

used are shown as tables accompanying figure 3.6, as well as the decision trees visualized in section A.2. Only the classifier responsible for *Bartonella* infection scoring could be constructed using a single set of boundaries, while yielding good results throughout all corresponding plates.

Adenovirus infection scoring is based on GFP signal intensity which varies across the cell body and concentrates within the nucleus. Strictly dependent on the amount of virus added to the cells, severity of infection is described well by signal intensity on the pathogen channel. The relevant features therefore include *Cells.Intensity_MeanIntensity_CorrPathogen*, *PeriNuclei.Intensity_MeanIntensity_CorrPathogen* and *Nuclei.Intensity_MeanIntensity_CorrPathogen*. The corresponding decision tree is shown in figure 3.6.

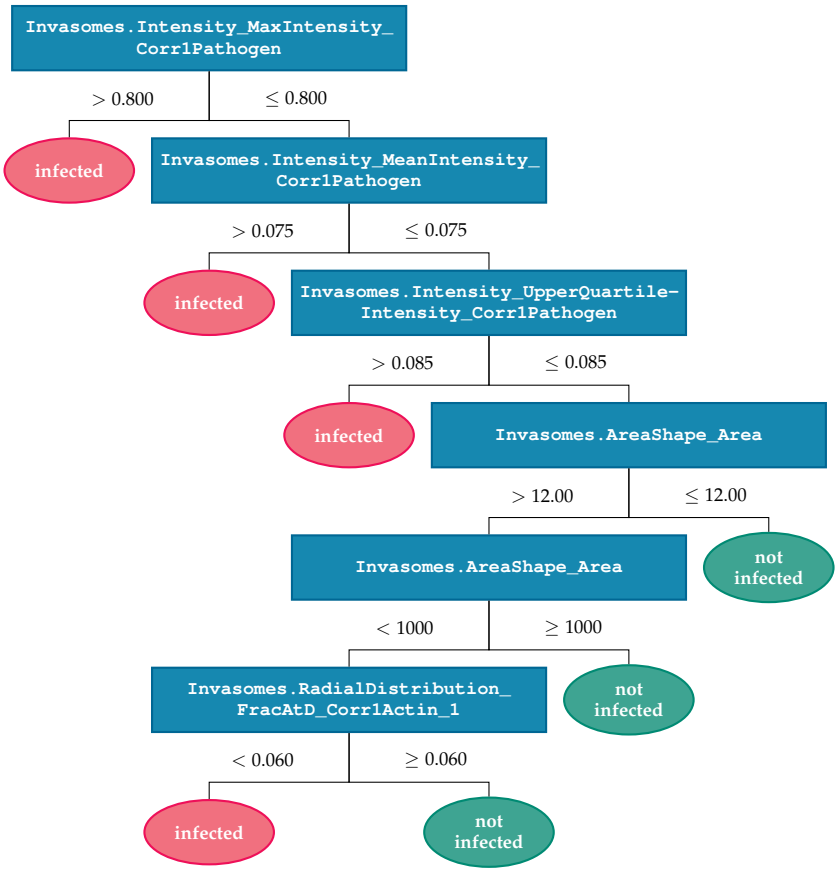


Figure 3.7: Decision tree for *Bartonella* infection scoring. In order to detect bona-fide invasomes, the first three or-linked decisions assemble a list of candidates while the following three and-linked decisions discard some erroneously included instances. In order to obtain the desired cell-based infection score, invasomes are mapped to cellular objects in a subsequent step. For *Bartonella*, a single set of decision boundaries was found to provide satisfactory results throughout all analyzed plates.

One mode of pathogen entry in *Bartonella* infection is via specialized invasome structures which can be described by an actin ring surrounding a large bacterial aggregate. These structures are detected on the actin channel, currently with a fair amount of false positives. Therefore, the pathogen channel is also taken into account by thresholding maximum, mean and upper quartile pathogen intensities within all invasome structures while remaining false positives are excluded based on size and radial distribution of actin intensity. Cells that contain one or more invasome structures that meet the criteria depicted in figure 3.7 are considered infected.

The remaining decision trees are shown in section A.2 but will be described here[4]. *Brucella* infection scoring is based on GFP mean intensities across the objects *Nuclei*, *PeriNuclei* and *Cell* body, which captures the typical infection pattern of large micro-colonies spread throughout the cell. In *Listeria* on the other hand, pathogen presence is evaluated though measuring bacterially secreted InlC on the Cy3 channel. The employed proxy depends on the amount of pathogen that is present and accumulates in the perinuclear region. Measurements used are mean nuclear intensity, mean perinuclear intensity and nuclear upper quartile intensity. The infection pattern for rhinovirus is characterized by cytoplasmic viral factories and virions are identified though stained anti-mouse immunoglobulin G, measured on the GFP channel. Targeted features are mean upper ten percent intensities in nuclei, perinuclei and Voronoi cells.

DTIS in the remaining two pathogens is even simpler, requiring only two features each. *Salmonella* screens depend on *SubCellBacteria* object segmentation, for which mean intensity and area features are relevant. The strain used for screening contains a GFP expressing plasmid that is controlled by an SPI-2 (SsaG)-dependent promoter. Therefore only intracellular bacteria are measured, allowing for reliable infection scoring. Finally, for vaccinia virus, pathogen intensity is measured on the GFP channel and a primary infection pattern is distinguished form a secondary. The former focuses on mean values in nuclear and perinuclear objects, while the latter additionally takes cells and Voronoi cells into account. As only primary infections are of interest within this project, solely the one corresponding decision tree is shown in section A.2.

[4] An exception in the employed infection scoring procedure is currently made for *Shigella* screens. The chosen strand is incapable of ABM (Δ*vir*G), leading to accumulation of micro-colonies in nuclear vicinity. Expression of pathogen marker DsRed, a form of red fluorescent protein (RFP), only occurs intracellularly, therefore alleviating the problem of including extracellular objects. This makes it possible to only rely on the pathogen channel in perinuclear objects and segment bacterial objects through Otsu's method.

Chapter 4

R Package singleCellFeatures

The format in which CellProfiler feature data is stored is only of limited suitability for exploratory data analysis. CellProfiler was originally implemented in the proprietary MATLAB language but has recently been ported to Python as version 2.x, in order to move away from the drawbacks of relying on a closed-source, commercial interpreter. Unfortunately, InfectX workflows are all based on CellProfiler 1.x and there are no current plans for updating to version 2.x. Consequently, all available single cell feature data is stored as MATLAB (Level 5) MAT-files.

The way in which storage is organized, while apt for working with a limited number of features corresponding to an entire plate, is unfitting if a large number of features belonging only to a subset of cells (e.g. all cells in a specific well) are of interest. Features are saved plate-wise in individual, gzip-compressed files, typically 1–3 MB in size and making the data contained in several hundred (depending on pathogen and generation the analysis pipeline, 500–700 features exist) such files available to an R session[1], using R.matlab version 3.2.0, Bengtsson, Jacobson, and Riedy (2015), takes on the order of 30 min.

As MATLAB does not constitute a tool that is particularly popular in the field of statistics and does not provide many of the convenience functions, available to R, that are much appreciated in exploratory data analysis, it was decided to convert single cell feature data as generated by CellProfiler 1.x into a format natively accessible by an R environment. Due to the amount of time involved,

[1] Using R version 3.2.0 (R Core Team 2015), installed as precompiled binaries running under Mac OS 10.10.5 on a 3.4 GHz Intel Core i7-2600 platform (iMac12,2) with 32 GB RAM. Whenever computational timing information is given and nothing else is specified, this is the reference system used to obtain the measurements.

this cannot be performed as a first step of every analysis and owing to the amount of storage necessary, it makes little sense to be carried out beforehand for all plates. Therefore, a system is needed, capable of fetching data that is not available locally, preprocessing it for direct access by R and storing the results for future use.

Furthermore, data-structures were developed, representing the hierarchy of single cell HTS data and capable of accommodating some associated metadata. Methods for operations that are frequently performed on such data are implemented in order to simplify many analysis tasks. With growing complexity of the code-base, it was decided to create an R-package that bundles the described capabilities.

Two similar projects, cellHTS2 (Boutros, Brás, and Huber 2006) and RNAither (Rieber et al. 2009), both hosted on Bioconductor (Huber et al. 2015), were looked at but none of them fulfilled the requirements imposed by the InfectX datasets. While cellHTS2 is designed for microarray data or siRNA data obtained by a plate reader (yielding a scalar value per well), RNAither can handle data at the single cell level. It is, however, geared towards running analysis on a single feature, obtained on a single imaging channel and cannot accommodate the heterogeneity of data available from the InfectX image analysis pipeline. In addition, RNAither is neither optimized for the large amount of data associated with several hundred features, nor does it provide the sought after tools for handling such a dataset, rather than implementing a fixed analysis procedure that can be readily applied to a single intensity feature. The newly developed singleCellFeatures therefore constitutes a further step in the evolution of R packages for siRNA data analysis, starting with cellHTS2 which is generalized in a vertical fashion by RNAither with the increase in resolution from wells to cells, which in turn is extended horizontally by singleCellFeatures to include many different features.

Much effort during development of singleCellFeatures was spent for ensuring the necessary flexibility to accommodate any possible kind of feature and for implementing some crucial sections in a way that is efficient enough for interactive usage. The former task is achieved by allowing features to consist of a single value per well, a single value per cell or a vector of values per cell and only minimally relying feature naming conventions, while the latter issue is best illustrated by the following introductory example.

As proposed by Knapp et al. (2011) and Snijder et al. (2012), the population context of each cell may significantly influence some morphological properties, as well as confound phenotypic information that is measured during feature extraction and therefore has to be accounted for. They propose several features that may act as proxies to characterize aspects of population context, one of

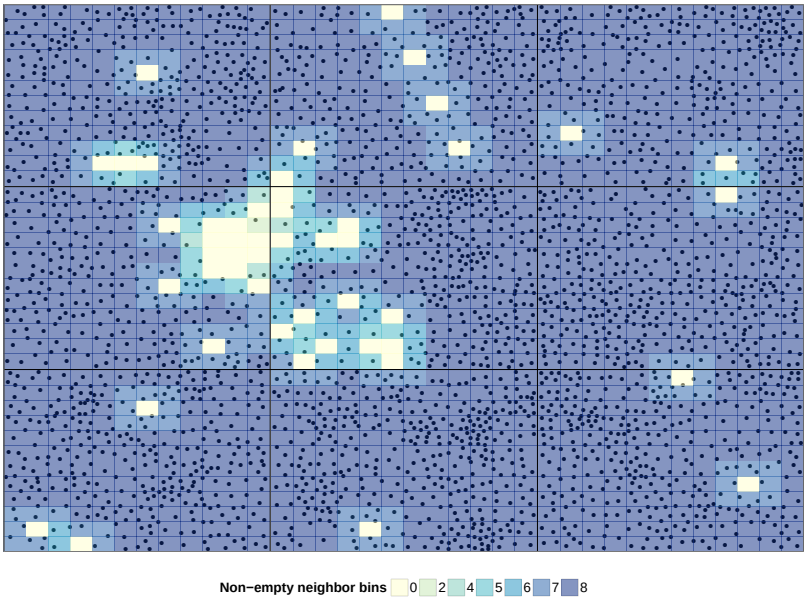


Figure 4.1: Cell colony edges are detected by 2D binning of cell center locations. Dots represent cell centers within the well H6 of plate J107-2C. Each of the nine images is segmented into 12 horizontal and 12 vertical sections yielding 144 tiles (1296 bins for the entire well). The tiles are colored according to the number of non-empty neighboring bins.

which is whether a cell is located towards the border of a colony or is surrounded by cells in all directions. In order to approximate this information from location data, images are divided into 2-dimensional bins or facets and the number of cells per bin is counted. Cells that are located adjacent to one or more bins that are empty are considered edge cells and cells surrounded by non-empty bins are center cells. Figure 4.1 visualizes the concept by color-coding facets according to the number of non-empty neighbors.

Code listing 4.1 constitutes an adaption of the implementation developed by Knapp et al., which is kindly provided as supplement to their publication. While iterating over all facets, only exploiting vectorization within facets and heavily relying on if-else logic, might be feasible when using datasets of the size the authors provide as demonstration material, combined with permanently storing the results, such an approach is impractical with datasets as produced by InfectX. Still, even for their datasets, the authors warn that:

Listing 4.1: In order to detect whether a cell is located towards the border of a colony or is surrounded by neighboring cells, each well is divided into 2-dimensional bins and the number of cells per bin is counted. As implemented by Knapp et al., all bins are iterated, each of the 8 possible directions (2 vertical, 2 horizontal and 4 diagonal) is checked for an empty neighbor and the corresponding binary value (at least one neighbor is empty) is saved to the current group of cells.

```
edgepos <- function(x, y, img, n) {
  empty <- logical()
  xst <- img[1] / n
  yst <- img[2] / n
  sgrid <- matrix(0, nrow=n, ncol=n)
  for (i in 1:n) {
    for (j in 1:n) {
      ispos <- (x > (i - 1) * xst) & (x <= (i * xst)) &
               (y > (j - 1) * yst) & (y <= (j * yst))
      sgrid[i, j] <- sum(ispos)
    }
  }

  for (i in 1:n) {
    for (j in 1:n) {
      ispos <- (x > (i - 1) * xst) & (x <= (i * xst)) &
               (y > (j - 1) * yst) & (y <= (j * yst))
      isempty <- FALSE
      if ((i > 1) && (j > 1) && (sgrid[i - 1, j - 1] == 0))
        isempty <- TRUE
      else if ((i > 1) && (sgrid[i - 1, j] == 0))
        isempty <- TRUE
      else if ((i > 1) && (j < n) && (sgrid[i - 1, j + 1] == 0))
        isempty <- TRUE
      else if ((j > 1) && (sgrid[i, j - 1] == 0))
        isempty <- TRUE
      else if ((j < n) && (sgrid[i, j + 1] == 0))
        isempty <- TRUE
      else if ((i < n) && (j > 1) && (sgrid[i + 1, j - 1] == 0))
        isempty <- TRUE
      else if ((i < n) && (sgrid[i + 1, j] == 0))
        isempty <- TRUE
      else if ((i < n) && (j < n) && (sgrid[i + 1, j + 1] == 0))
        isempty <- TRUE
      empty[ispos] <- isempty
    }
  }
  return(empty)
}
```

> These [population context feature] computations require considerable amounts of memory, and will take some time. This must be done for each of the input files, and will produce an output file containing the input data plus computed population features.

An execution of the code, using the supplied exemplary dataset (number of cells per well: $\mu = 464$, $\sigma = 168$; 15 bins in x-direction and 15 bins in y-direction) reveals that of the 2899.83 s spent on calculating population context features, 2892.67 s (99.8% of the total time) is spent on determining positions within colonies. This performance can be expected to deteriorate significantly, using InfectX data, due to a ten-fold increase in cell counts and more importantly a 3-fold rise in resolution in both x- and y direction. The surge in dataset size entails an increase in number of bins needed, to obtain sensible results (the example in figure 4.1 is divided into 36 bins per dimension) and runtime scales as $\mathcal{O}(n^2)$ in number of bins per dimension.

Neither the high computational cost, and consequently not even storing the results, however are necessities, as demonstrated by a slightly optimized implementation (see listing 4.2). Fully vectorizing the problem no longer requires the extensive use of if-else logic and due to R being an interpreted language, built for operating on vectors, such an approach can be expected to bring about compelling speed improvements. Vectorization is achieved by calculating the indices of the 8 neighboring bins for each facet and applying this stencil to the 2-dimensional matrix containing binary empty/non-empty information per bin. This efficiently yields the number of empty neighbors for each facet which then can be mapped to cells by their column-major grid indices. An additional advantage of the presented implementation is that the number of empty neighbors is obtained for free whereas the code proposed by Knapp et al. would perform even worse if modified to produce this supplementary information, as the full set of if-statements has to be run to the end for each facet.

Benchmarking the two implementations reveals that a 66-fold speedup is easily achievable only by completely vectorizing the problem. While the original code takes 198.08 s per plate (384 iterations of the same well; $\mu = 515.85$ ms, $\sigma = 14.66$ ms; well H6 of plate J107-2C), total runtime is reduced to 2.98 s ($\mu = 7.77$ ms, $\sigma = 2.69$ ms, per well).

Such time-savings might not look that significant at first sight. But within the context of the available data, consequences of critical optimizations are of great value. First of all, interactive data analysis is sensitive to waiting times that exceed a few minutes and due to there not only being a single location feature to be analyzed, but rather 5–10, several minutes per feature are impractical. Moreover, if run-times can be sufficiently reduced, this makes it possible to run calculations on the fly instead of constantly saving results. Apart from

Listing 4.2: Due to the larger number of cells per well and the increase in image resolution as compared to data used in Knapp et al. (2011), calculation of border/center population context features have to be carried out in a more efficient manner, which can be accomplished by fully vectorizing the problem.

```
facetBorder <- function(x, y, img, facet) {
  facet.size <- img / facet
  # calculate facets (2d binning)
  x.bin <- ceiling(x / facet.size[1])
  y.bin <- ceiling(y / facet.size[2])
  # initialize empty grid/border matrices
  grid <- matrix(0, facet[2], facet[1])
  # calculate col-major grid index for each object
  index <- y.bin + (x.bin - 1) * facet[2]
  # summarize as counts
  counts <- table(index)
  # fill grid with counts
  grid[as.numeric(names(counts))] <- counts
  grid.res <- grid
  grid <- grid > 0
  # extend grid with a frame of ones
  grid.ext <- rbind(rep(1, (facet[1] + 2)),
                    cbind(rep(1, facet[2]), grid, rep(1, facet[2])),
                    rep(1, (facet[1] + 2)))
  # set up stencil
  row <- rep(rep(1:facet[2], facet[1]))
  col <- rep(1:facet[1], each=facet[2])
  colP1 <- col + 1
  colM1 <- col - 1
  rowP1 <- row + 1
  rowP2 <- row + 2
  nrowP <- facet[2] + 2
  stencil <- cbind(row   + colM1 * nrowP, # northwest neighbor
                   row   + col   * nrowP, # north neighbor
                   row   + colP1 * nrowP, # northeast neighbor
                   rowP1 + colP1 * nrowP, # west neighbor
                   rowP2 + colP1 * nrowP, # east neighbor
                   rowP2 + col   * nrowP, # southwest neighbor
                   rowP2 + colM1 * nrowP, # south neighbor
                   rowP1 + colM1 * nrowP) # southeast neighbor
  # apply stencil row-wise to grid
  border <- apply(stencil, 1, function(ind, mat) {
    return(sum(mat[ind]))
  }, as.numeric(grid.ext))
  # map col-major object index to border array
  return(border[index])
}
```

the obvious downsides arising from storage overhead, management of result caches leads to practical issues whenever the workflow is under development and thus subject to constant change. Caches have to be invalidated and recomputed and the system overseeing these processes has to be made aware of changes, as well as understand where they apply.[2] Furthermore, updating batches of data at once incurs substantial cost which might even be expended in vain if not all data is used before the next update triggers recomputation. Just in time generation of derived features is preferable over ahead of time calculation and the benefits clearly justify some optimization efforts.

When looking at runtimes of the original implementation modified for benchmarking on InfectX data (198.08 s) and for a single execution of untouched, original code on the developer-supplied exemplary dataset (2892.67 s) side by side, a large discrepancy stands out. The two values are not directly comparable, as they are created with individual binning and differently sized datasets. But instead of explaining the divergence, these factors separate the two values even further, as the benchmarked time was obtained using a dataset roughly ten times larger and taking into account 6-fold more bins (225 versus 1296). Running the code of Knapp et al. with InfectX data, as to be expected, takes significantly longer and determining cell position within colony for a single location feature over an entire plate is estimated to run for 41.14 h.[3] The exact same results can be computed within 8.04 s by the code integrated in singleCellFeatures through vectorization as discussed previously and by splitting the data into wells.

Listing 4.1 does not entirely reproduce the code of Knapp et al., but constitutes a simplified version intended for highlighting the importance of vectorization and is adapted for comparison with the optimized variant. Originally, the nested for-loops iterating the bins (lines 6–12 and 14–37), are enclosed by an additional for-loop, iterating the wells. Data is stored as a large `data.frame` and all operations are performed on full-length vectors. The enormous number of comparisons resulting from this setup leads to serious performance issues, which are amplified when increasing the number of cells from ~180000 to ~10^6 per plate. Such observations, along with other considerations lead to development of a more sophisticated data structure for storing single cell feature

[2] On an anecdotal note, Philip Karlton, a software engineer and former Netscape executive is quoted as saying: "There are only two hard things in Computer Science: cache invalidation and naming things," corroborating the claim that it is desirable to avoid caching altogether whenever possible.

[3] Due to the long running time, not all wells are processed, but 10 wells are randomly sampled and the mean per well is extrapolated to the full plate. The used wells are A1, A6, B1, B23, B9, E11, E14, I4, N20, P2 of plate J107-2C. Moreover, it has to be emphasized once again, that the reported timespan corresponds only to the calculation of a single location feature, several of which are typically available.

data (see section 4.1), which splits the data into smaller units and consequently eliminates constant comparisons against well indices for per-well operations.

The following sections discuss some design aspects and implementation issues of singleCellFeatures, beginning with data structures developed for representing single cell feature data, continuing with how data is acquired and locally cached, management of metadata and concluding with some tools for manipulating datasets as well as providing the data to third-party tools. This architectural overview is complemented by a more practical section in appendix B. For all coding, the style guide outlined in Wickham (2014) was followed and Wickham (2015) served as an invaluable reference, along with the extensive documentation supplied by R Core Team (2015). Package documentation is in-source with `.Rd` files generated by roxygen2 (Wickham, Danenberg, and Eugster 2015), which are accessible from within the R help system upon installation of the package.

4.1 S3 Classes

Object oriented programming (OOP) in R can be achieved by any one of the three distinct object systems S3, S4 and reference classes (RCs). While the RC system resembles the OOP style most people coming from languages such as Java or C++ are familiar with, by implementing message-passing, some features such as mutability (in-place modification of objects) and pass-by-reference semantics violate common expectations of R users. S3 and S4 objects are built on the concept of generic function OOP, which does not associate methods with classes, but employs a special type of function, called a generic function, that is responsible for method dispatch. Of the two, S3 classes are more loosely organized, lacking the formal definitions of the S4 system which can describe both representation and (multiple) inheritance. Furthermore, S4 objects are capable of multiple dispatch and all code used for creating and manipulating S4 objects is not part of base R but supplied by the methods package, which introduces the new `@` operator, for accessing object slots.

Much base functionality in R is implemented using S3 classes, including lm and glm objects, the support for which has been available to R from its very beginning. The methods package for S4 objects is attached by default since version 1.7.0, but fewer packages make use of the newer syntax, some examples being the base package stats4 and CRAN packages Matrix and lme4. Bioconductor packages on the other hand are frequently designed on top of S4 classes. RC was only introduced with R 2.12 as part of the methods package and therefore currently does not enjoy widespread adoption. The limitations of S3 were found to be unproblematic for the planned feature set of singleCellFeatures and

due to the attractive simplicity of this scheme, several objects are implemented as S3 classes.

4.1.1 Metadata Objects

Metadata is extracted from aggregate files (see section B.3.3) and used to generate both plate-level and well-level metadata objects. These consist of key-value pairs stored as lists and accompany all plate-level and well-level data objects generated by singleCellFeatures. The R class attribute names are `PlateMetadata` or `WellMetadata` and both are associated with a superordinate object name `Metadata` in order to simplify method dispatch in cases where a function is able to operate on both object types. Table 4.1 shows the 12 slots along with short descriptions that constitute a `PlateMetadata` object, while table 4.2 does the same for objects of type `WellMetadata` (20 key-value pairs in total). Several slots, namely `plate.barcode`, `plate.quality`, `experiment.name`, `experiment.pathogen`, `experiment.geneset` and `experiment.library`, are redundant and therefore only listed in table 4.1.

Having metadata structures available alongside the objects holding single cell features facilitates manipulation of the data as all relevant information is bundled and does not have to be managed separately by the user. Furthermore, information stored in `well.type` can be used for normalization (sometimes it is desirable to only normalize non-control wells) and fields like `gene.name`, `gene.id` or `sirna.name` are used often for selecting the desired subset of data. The two slots `plate.quality` and especially `well.quality` hold promise for automatically discarding bad data, but current annotation levels leave much to be desired (87% and 92%, respectively, are labeled as `UNKNOWN`) due to the large amount of manual labor that is involved. Perhaps, if some form of automatic quality assessment is developed or if the results from focus detection are utilized to enhance data quality comments, these fields could be put to better use.

Originally, inspired by the cellHTS2 package, it was planned to use public databases for collecting further metadata, such as gene ontology, chromosomal location, gene function summaries and homology. The bioconductor package biomaRt (Durinck et al. 2005, 2009) can be used to query several web services, including Ensembl, Uniprot and Reactome with gene IDs (Maglott et al. 2011) and retrieve gene annotations. Furthermore, siRNA sequences could be used for investigating OTEs and developing methods in order to correct for corresponding effects. Unfortunately, time constrains so far have prevented such ideas from being explored further.

Table 4.1: `PlateMetadata` structures consist of 12 key-value pairs intended to capture all relevant plate-wide metadata.

Key name	Description
`plate.barcode`	The unique identifier assigned to every plate, for example `KB02-1L`.
`plate.quality`	Plate quality descriptors are `UNKNOWN`, `GOOD`, `BAD` and `WARNING`, but currently most (87%) are associated with the label `UNKNOWN`.
`plate.type`	Possible values for plate type are `ScreeningPlate`, `CheckerBoard` and `MockPlate` and different types are characterized by their control layout.
`experiment.space`	The first hierarchy level of data organization in openBIS. Current spaces are `INFECTX` and `INFECTX_PUBLISHED`
`experiment.group`	The group level is used to sort data according to pathogen (e.g. `ADENO_TEAM`).
`experiment.name`	At the experiment level, data is grouped into screens. An example value is `ADENO-DP-K1`.
`experiment.pathogen`	While the pathogen is already encoded in the openBIS hierarchy of InfectX, this can also be handled differently. In case of an alternative setup, the treatment applied to the screen can be stored separately.
`experiment.geneset`	Different sets of genes are investigated throughout all screens and possible values are `Drug`, `Validation`, `MicroRNAome`, `Genome` and `Kinome`.
`experiment.replicate`	Designates the replicate number of the current plate (see table 3.1).
`experiment.library`	Manufacturers that provide siRNA reagents to InfectX are `Selleck`, `Ambion`, `Sigma`, `Dharmacon` and `Qiagen`.
`experiment.batch`	Alphanumeric identifier for the batch in which the current plate was put through the wet-lab procedure.
`counts.quantiles`	The lower and upper 5% quantiles for cell counts over all 3456 images of the plate, determined in order to detect bad wells.

4.1.2 Data Objects

Central to the presented R package are data structures that hold single cell feature data. Instead of just storing all data as a large `data.frame` as in RNAither, a more complex scheme was developed that represents the physical hierarchy of HTS datasets. This has both the advantage of presenting an intuitive structure of the data that can easily be navigated by the user, as well as efficiency benefits. Selecting cells from a well does not involve 10^6 integer comparisons, but finding a slot in a list structure of length 384 (for consequences of the former approach, see the introductory example staring on page 84).

Three types S3 objects represent the levels, data can be sorted by: 6 or 9 `ImageData` classes are assembled into a `WellData` object (which is additionally associated with a `WellMetadata` object) and 384 `WellData` structures, together with a `PlateMetadata` instance, compose a `PlateData` object. The respective

Table 4.2: Analogously to `PlateMetadata` objects, `WellMetadata` classes consist of several key-value pairs. All slots that appear in both `Metadata` definitions (`plate.barcode`, `plate.quality`, `experiment.name`, `experiment.pathogen`, `experiment.geneset` and `experiment.library`) are excluded from this overview. Please refer to table 4.1 for more information.

Key name	Description
`well.row`	Capitalized alphabetic index of the current well row ($\in \{A, B, \ldots, P\}$).
`well.col`	Integer-valued column index of the current well ($\in \{1, 2, \ldots, 24\}$).
`well.index`	Well indices are calculated from row and column location and represent linearized, row-major well positions within plates ($\in \{1, 2, \ldots, 384\}$).
`well.type`	Descriptor for the type of well, the most important of which include `SIRNA`, `POOLED_SIRNA` and `CONTROL`.
`well.quality`	Possible values for this field are `UNKNOWN`, `WARNING` and `BAD`, while most wells currently are set to `UNKNOWN` (92%).
`gene.name`	Gene symbol corresponding to the targeted gene (Gray et al. 2013).
`gene.id`	Entrez Gene ID of the targeted gene (Maglott et al. 2011).
`sirna.name`	The siRNA catalog ID as specified by the manufacturer.
`sirna.sequence`	Full sequence of the 5′-3′ siRNA antisense (guide) strand added to the current well.
`sirna.seed`	The seed sequence (nucleotides 2–9 from the 5′ end) of the 5′-3′ siRNA guide strand.
`sirna.target`	Sense sequence of the siRNA target.
`counts.cells`	The cell count of the current well.
`counts.pathogen`	Count of recognized pathogen objects (in *Bartonella* screens, the number of invasomes).
`counts.infection`	Number of infected cells according to DTIS.

collections of child objects are gathered in list structures named `data` (figure 4.2 presents a visualization of this hierarchy). Name attributes are organized similarly as for metadata objects, with a superordinate tag `Data` assigned in addition to the individual descriptors, in order to be able to define functions that can be applied to all data objects in the same way.

The next logical level is the screen (siRNA library), but clustering several `PlateData` objects is infeasible with current hardware. As R keeps all loaded objects in memory and a plate consisting of ~600 features on $\sim 10^6$ cells requires ~8 GB, only large memory systems could handle such data structures.[4]

[4] A large memory Euler node (a centralized HPC resource provided by the ETH Zürich) was successfully used to operate on a collection of 8 `PlateData` objects for screen wide normalization trials, but the required ~220 GB of memory are currently not commonly available to desktop environments. Simply keeping the data in memory is not all that is required, as algorithmic overhead has to be factored in as well.

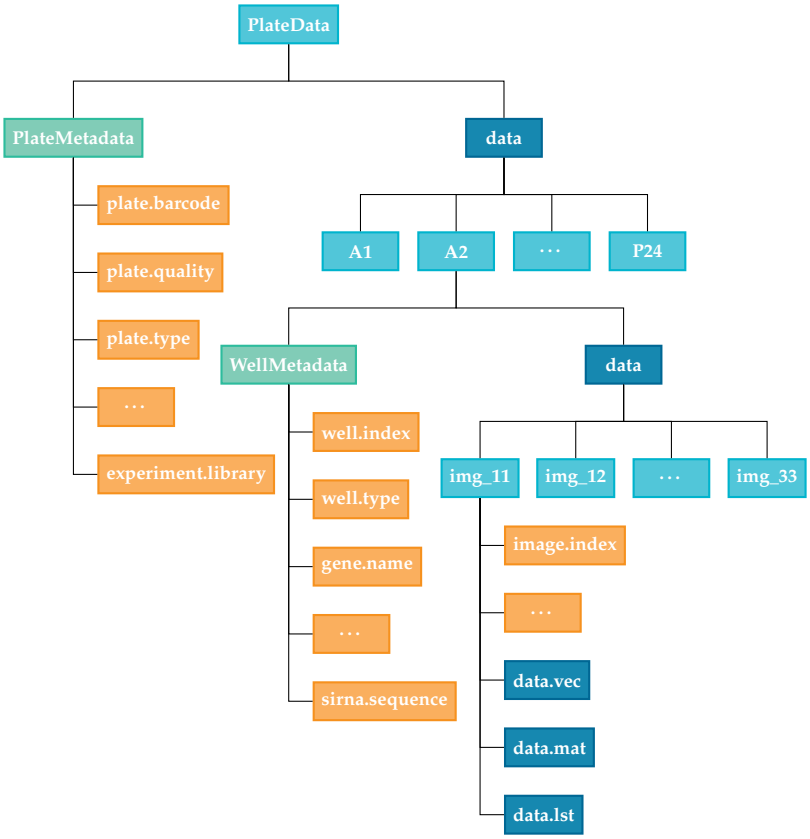


Figure 4.2: Single cell data is organized according to the hierarchy imposed by HTS experiment design. Plates are divided into wells which are subdivided into images and Metadata objects (green) are attached at each level. Data objects provided by singleCellFeatures are in light blue and contain lists of subordinate objects (dark blue), while Metadata objects consist of key-value pairs (orange).

The slots of `ImageData` objects are `plate` (barcode), `well.index` (linear, row-major and $\in \{1, 2, \ldots, 384\}$), `well.row` ($\in \{A, B, \ldots, P\}$), `well.col` ($\in \{1, 2, \ldots, 24\}$), `image.index` (linear, row-major and $\in \{1, 2, \ldots, 6\} \vee \{1, 2, \ldots, 9\}$, depending on number of imaging sites), `image.row` ($\in \{1, 2\} \vee \{1, 2, 3\}$, for 6 and 9 image wells, respectively), `image.col` ($\in \{1, 2, 3\}$, independent on total number of images), `image.total` ($\in \{6, 9\}$), `counts.quantiles` (the lower and upper 5% quantiles of plate-wide, image level cell counts), `counts.cells`, `data.vec`, `data.mat` and `data.lst`.

Actual feature data is sorted according to dimensionality with respect to images or detected objects within images. Scalar-valued entities (such as object counts, maxima, minima, etc.) are saved into `data.vec` which represents the concatenated data as a vector-valued `data.frame` (in order to accommodate both numerical and character fields). Vector-valued features (the bulk of single cell data) are assembled into matrices, split by object counts, which are added to `data.mat`. While there are typically as many nuclei as cells, those features can be merged into the same matrix, but pathogen objects typically require their own data structure. Furthermore, it may happen that cellular features differ in length due to segmentation errors. A final group of features is vector-valued per object (matrix-valued per image) and is saved to the `data.lst` node. Currently, neighbor features are detected and converted to sparse adjacency matrices, using the Matrix package (Bates and Maechler 2015), while others, such as parent/child features, are represented by nested list structures.

One obvious advantage of the proposed data representation is storage efficiency for values that apply to groups of cells, such as plate and well-level metadata or all features of the `data.vec` group. When using a large matrix for all data, this information has to be duplicated $\sim$400 times at image level, $\sim$3500 times at well level and $\sim 10^6$ times at plate level, incurring significant storage overhead. This can be mitigated somewhat by using factors and the resulting structures can still be handled well by current hardware, but nonetheless, parsimony is desirable. For these reasons, metadata objects at image level were not developed, as they unnecessarily inflate object size, despite this breaking recursive symmetry. Instead, only the minimally necessary information for identifying the data source is attached at this level.

In retrospect, it is debatable whether spiting data into images is sensible or creates unnecessary computational overhead in traversing nested lists. At the time of original implementation it was unclear if all images could be treated equally and having this level of granularity available seemed important. With access to some practical experience in working with such structures, this feature may not be worth the complexity it entails. Splitting the data at well level, however has proven to be worthwhile.

Listing 4.3: Mostly used for internal communication of dataset identity and location both within openBIS and the local file system, `DataLocation` objects simplify certain tasks such as lookup of cache files.

```
getCacheFilenameData <- function(x) {
  UseMethod("getCacheFilenameData", x)
}

getCacheFilenameData.WellLocation <- function(x) {
  config <- configGet()
  name <- paste0(x$plate, "_", x$row, x$col, "_sc_all.rds")
  path <- paste(config$dataStorage$dataDir, x$space, x$group,
    x$experiment, x$plate, "WellData", name, sep = "/")
  return(path)
}

getCacheFilenameData.PlateLocation <- function(x) {
  config <- configGet()
  name <- paste0(x$plate, "_sc_all.rds")
  path <- paste(config$dataStorage$dataDir, x$space, x$group,
    x$experiment, x$plate, name, sep = "/")
  return(path)
}

getCacheFilenameData.default <- function(x) {
  stop("can only deal with DataLocation (PlateLocation/WellLocation) objects.")
}
```

4.1.3 Auxiliary Objects

Several additional S3 objects are implemented in an auxiliary capacity. Both `PlateLocation` and `WellLocation` classes serve to communicate the identity and location of datasets, while `PlateAggregate` and `MatData` objects are used for caching purposes.

Objects representing data locations are simple key-value lists that are instantiated with a barcode in case of `PlateLocation` or a barcode and a row/column specification for `WellLocation` objects. Subsequent population of slots `space`, `group` and `experiment` is performed by lookup of the information in the `plate.database` table (see section B.3.1), which is an optimization measure necessary to speed up creation times severely hampered by retrieving the values from large aggregate files. The information contained in `DataLocation` (the class attribute to specify any of the two location types) objects uniquely specifies where the described dataset is located on openBIS and in local filesystem cache and these classes are mainly used for unifying package-internal communication.

Before introduction of `DataLocation` objects, a fair amount of redundant code was necessary for simple tasks such as saving or retrieving cache files. Identity of datasets was mostly specified by passing plate barcodes/experiment names and from that information, paths were assembled. Cumbersome casting of letter case, splitting strings to separate pathogen from library and adding setup-specific strings (such as `_TEAM`, which added as suffix to an all caps pathogen name yields the openBIS project name) to generate the appropriate values determining local file paths, as well as performance issues due to constant lookups of the same information can altogether be avoided by using specialized objects containing all necessary information. Furthermore, the S3 method dispatch system can be put to use for specifying different behavior dependent on the object that is used as function argument. For a small illustration of such as use case, see code listing 4.3.

The two remaining auxiliary objects are both used for caching purposes. Due to the narrow purpose of `DataLocation` a lookup table containing all required information can be created, that remains small enough to be loaded quickly. This is not possible for creating metadata objects, as most aggregate file columns are required for this process. In order to prevent repeated loading of these large files (100–200 MB, plate-level excerpts are cached as `PlateAggregate` objects whenever first accessed. The resulting files are small (~100 kB) and can be attached quickly. The data structure is simple, containing of two slots `plate` and `data`, the former holding a `PlateLocation` object and the latter a `data.frame` storing the relevant 384 rows of the corresponding aggregate file.

Finally, `MatData` is a type of `Data` object, that can be thought of as a precursor to a `PlateData` object. The reason for its existence lies in the dynamic nature of this project. Many parts of singleCellFeatures evolved considerably and constant change poses problems for cached data structures. Originally, `PlateData` objects themselves were saved to the local file system but whenever their implementation was modified, all instances had to be invalidated and rebuilt (which initially took ~45 min per plate). In order to improve the situation, not `PlateData` objects are written to disk, but a data structure that holds all MATLAB feature data in a large nested list, as it is imported into R. Two slots are available to a `MatData` object, one called `meta` and intended for holding a `PlateMetadata` object and the other is named `data` and stores all single cell features.

This successfully addresses the outlined problem, since the raw data will not change frequently (it still can occur that for example new features are developed[5], or better segmentation is employed), but introduces a new issue: build up of `PlateData` objects from `MatData` has to be reasonably fast due to this

[5]The system is built with a fair amount of flexibility, as there are methods available for adding

being the first step in every interactive session. Some aspects of how this is achieved is discussed in section 4.2.

None of these auxiliary data structures are implemented as S3 objects for simply defining a set of fields, but much rather for their capability of using generic functions that specify how a specific procedure is applied to the different object variants. While this by itself is not an indispensable feature, as one could simple choose different but related names for the set of procedures, it is greatly appreciated for helping to organize code into functional units. Furthermore, this reduces the amount of redundant code by allowing the same function name to be applied in different contexts (for examples of this, see listings 4.6 and 4.8).

4.2 Data Access

The core capability of singleCellFeatures is seamlessly providing the user with single cell data and doing this within time-frames that are suitable for interactive usage. As already discussed in the introduction to this chapter, some form of local caching is necessary to meet these goals, as downloading data from openBIS and especially loading a large number of MATLAB `.mat` files into R takes a considerable amount of time.

Several tools were developed, beginning with search capabilities in order to locate the datasets of interest, followed by a set of functions that sort the queried data into units such that fetching is as economical as possible thusly minimizes waiting times and trigger retrieval from a multilevel cache system. All of this is handled in the background with no user intervention required. The following short example is intended for highlighting the effectiveness of the proposed system.

```
time <- system.time({
  plates <- findPlates(contents="MTOR",
                       experiments="brucella-du-k[12]")
  wells  <- findWells(contents=c("MTOR", "SCRAMBLED"),
                      plates=unlist(plates))
  data   <- getSingleCellData(wells)})
```

In order to retrieve data from all plates containing wells with siRNA directed against MTOR, a list of `PlateLocations` is generated by `findPlates`. Acting on that structure, `findWells` selects all wells containing scrambled control

new features to existing `MatData` structures, including automatic invalidation of downstream well caches. Therefore, extraction of new features does not require the rebuild of whole `MatData` objects.

siRNA, as well as MTOR knockdown wells and returns a list of `WellLocations`. Retrieving the resulting 104 wells, spread over 8 plates (1.22 GB of data), is accomplished within 44 s. In this case, all well data was read from cache, but apart from an increase in runtime, the same code snippet can be used irrespective of local cache state and will produce identical results. Figuring out how to proceed most efficiently in any case is handled by singleCellFeatures.

The most important steps in this procedure are visualized by figure 4.3. For sake of brevity, several aspects that complicate the logic underlying data access are omitted from the diagram. Some examples include recovery from failed downloads or imports, checks for completeness and consistency of the fetched data, issues involving metadata and the ability to only retrieve a subset of single cell feature, which requires special attention to down-stream caching. Color coding corresponds to the functional units searching (green), ordering the query and executing individual fetches (blue), as well as building the required data structures (orange).

Searching relies on search indexes that are stored as a single plate database and pathogen specific well databases in order to be carried out quickly and economically. These lookup tables are derived from metadata contained in aggregate files and are described in detail in sections B.3.1 and B.3.3. Further information regarding search functionality, such as arguments recognized by the two functions and how exactly search strings are matched is available in section B.4. The resulting list of `DataLocation` objects is passed to the function `getSingleCellData`, which acts as a conduit between searches and fetches.

The inquiry has to be ordered plate wise, in case uncached wells are sought from different plates. If such requests are carried out interleaved, large plate caches are read redundantly, entailing unnecessary overhead. The plates involved are iterated and it is checked if only cached wells are requested, in which case they are read from disk and added to the result structure. If uncached wells need to be made available or a complete plate is solicited, the corresponding `PlateData` object has to be built, the particularities which (orange nodes in figure 4.3) will be explored throughout the following subsections. Finally, if only wells were requested, they are extracted from the `PlateData` structure and added to the list of results and if the complete plate data is of interest it is returned as-is.

Green and blue nodes in figure 4.3 can be bypassed entirely if only few single data structures are of interest. Instantiating a single `WellData` object, for example, that is not cached, will trigger the build up of a `PlateData` structure, followed by the extraction of that well. This, of course is inefficient, if multiple wells of the same plate are needed, as the plate structure is repeatedly created and discarded. In such a scenario, the functions described above can be uti-

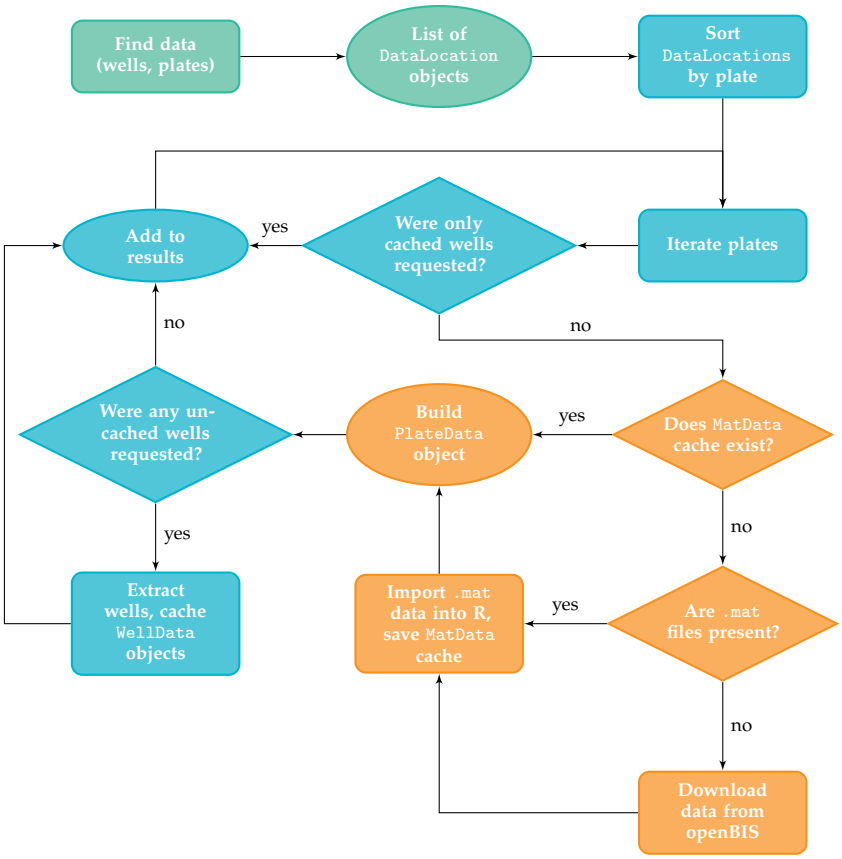


Figure 4.3: Three separate procedures are involved in loading single cell data. First, the functions `findPlates` and `findWells` can be used to run queries and retrieve a list of `DataLocation` objects (green) which is passed on to the function `getSingleCellData` (blue). Data requests are sorted according to plate and the list of plates is iterated. Dependent on request and availability of caches, `DataLocation` objects are assembled (orange) and returned to the calling function. This diagram represents a simplified view and several additional features complicate the underlying logic. For example, only a subset of features can be requested, in which case, writing of well caches has to be handled separately. Furthermore, checks for data completeness and consistency are employed that may alter the outlined execution path and mechanisms for recovery from unsuccessful imports are also in place but not shown.

lized, or PlateData object itself can be instantiated, following well extraction initiated by the user (via the functionality provided by extractWells). These concepts are best illustrated by a quick example:

```
# set up a few data location objects
well.locs <- list(WellLocation("J107-2C", "H", 6),
                  WellLocation("J107-2C", "G", 23))
# inefficient if well caches do not exist, as the corresponding plate
# cache will be loaded twice, otherwise quickest option
wells <- lapply(well.locs, WellData)
# best if plate data already loaded or well caches inexistent
plate <- PlateData(convertToPlateLocation(well.locs[[1]]))
wells <- extractWells(plate, well.locs, keep.plate=FALSE)
# singleCellFeatures figures out how to proceed optimally
wells <- getSingleCellData(well.locs)
```

The remainder of the current section will focus on topics revolving around data provisioning itself, involving data acquisition, caching and build-up of PlateData structures (orange nodes in figure 4.3).

4.2.1 Data Download

Fetching data from openBIS, in case neither corresponding MATLAB files, nor a MatData cache file exists,[6] is performed by issuing a system call and running the Java command line executable that is supplied with openBIS. Currently, three different types of files can be downloaded

HCS_ANALYSIS_CELL_FEATURES_CC_MAT
HCS_ANALYSIS_CELL_DECTREECLASSIFIER_MAT
HCS_ANALYSIS_CELL_THRESHOLDEDINFECTIONSCORING_MAT

and the downloading routine can be run in two different modes, one targeting single cell feature files (filtering for the first file type, FEATURES_CC), while the other is intended for fetching DTIS results and prefers DECTREECLASSIFIER but falls back to THRESHOLDEDINFECTIONSCORING when the former is unavailable. The two types of infection scores represent two generations of DTIS procedures (distinct from SVM based scoring which were used before) and while the newer files are available for many plates, there are some that have not yet

[6] A third scenario that will trigger data download is manual override. If, for some reason cached data gets corrupted, or if a newer version of a dataset is available on openBIS, the user can choose to re-download individual plates. In addition, removal of all caches can be triggered, but for most circumstances this presents too drastic a measure.

been updated. The downloaded set of files can further be refined by supplying a regular expression that is passed on the Java utility, but this will prevent downstream caches from being created (at well level) and create a plate level cache that is incomplete (can be completed without re-fetching the already downloaded data). All `.mat` files are saved to a temporary location within the directory structure managed by singleCellFeatures.

4.2.2 Data Import

Following completion of all download tasks, the resulting list of files is read into R using the function `readMat` of R.matlab (Bengtsson, Jacobson, and Riedy 2015). For unknown reasons this function sometimes fails to import a file,[7] which is caught, the affected file is saved into a special directory and import is allowed to continue. Due to this being the most time-consuming step in data provisioning, the results are stored as incomplete `MatData` caches and functionality for completing the data structure as opposed to entirely rebuilding it, are available. The function `rebuildCache` automatically retries files that failed during initial import and is capable of re-downloading individual features from openBIS that are missing from the `MatData` structure without user intervention.[8,9] Furthermore, the location of manually downloaded files can be specified in order for them to be added to the data structure and down stream caches of `WellData` structures are automatically removed whenever the `MatData` object is rewritten, as they are no longer valid.

Upon successful assembly of a `MatData` object, it is written to disk. Due to the large size of these objects (a typical plate is ~8 GB in size) good and efficient compression is essential both from a perspective of economically utilizing storage space, as well as from the viewpoint of time consumption. All major

[7] Failure is rare and does not seem to be dependent on file-level corruption, as retrying the same file usually resolves the problem. Furthermore, so far, no patterns could be made out, in that affected files appear to be somewhat random. The issue was not investigated further as a pragmatic approach resolved the problem.

[8] The necessity for re-downloading is distinct from re-importing. While the latter arises from the possibility of failed reads of `.mat` files, the former is a consequence of problems that surround the openBIS hardware implementation of InfectX. Due to infrastructure work during much of the time of this project, it could not be guaranteed that the downloaded set of features comprises the full set of available features. This issue is remedied by checking for completeness of the local feature-set after build-up of a `MatData` structure in combination with semi-autonomous and cheap addition of individual features in case some are found to be missing.

[9] Missingness currently is determined by comparison to a pathogen specific list of features. While being a very simple approach, it may lead to false positives (features that are indicated to be missing but do not exist), as well as false negatives (features that are present but indicated to be superfluous), as the set of features is not only pathogen specific, but also dependent on the analysis pipeline version. Improvements to this scheme are feasible (for example querying openBIS metadata directly or creating screen level lists) but not implemented due to time constraints and acceptable performance of the current system (see section B.3.2 for more information).

Listing 4.4: Compression with base library functions is not sufficiently performant, leading to the implementation of functions wrapping around `readRDS`/`saveRDS` that are capable of exploiting parallelized compression algorithms. The best overall compromise between compression/decompression speed and file size was obtained with pigz at compression level setting `-9`.

```
saveRDSMC <- function(object, file, threads = getNumCores()) {
  message("using ", threads, " threads for compression.")
  con <- pipe(paste0("pigz -p ", threads, " -9 -f > ", file), "wb")
  saveRDS(object, file = con)
  on.exit(if (exists("con")) close(con))
}

readRDSMC <- function(file) {
  con <- pipe(paste0("pigz -d -k -c ", file))
  object <- tryCatch({
    readRDS(file = con)
  }, error = function(err) {
    stop("could not read file\n", file, ":\n", err)
  }, finally = {
    if (exists("con")) close(con)
  })
  return(object)
}
```

compression programs (bzip2, gzip, xz and zip) offer a parameter to adjust the speed/file size trade-off (usually an integer between 1 and 9) but a high compression ratio entails significant performance penalties. For storage of `MatData` objects, smallest file sizes were obtained using xz with flags `-9` or `-e`, which were on the order of 1–1.5 GB.

While slightly longer compression times might be worth the savings in storage space, the time required for decompression is critical, as it will be the first step of every analysis procedure. Unfortunately, retrieval of highly compressed xz files is too slow for this application. Moreover, parallelized implementations lbzip2, p7zip, PBZIP2, pigz and xz with option `-T` were tested and the best overall compromise is achieved using pigz with option `-9`, as shown in code listing 4.4. Typical file sizes using this procedure are 1.5–2 GB for a complete `MatData` object and compression takes around 10 min.

After the plate cache file is written, all successfully read `.mat` files are deleted, while failed files are kept for subsequently re-attempting import. Whenever the cache is read from disk, it is checked for feature completeness and the user is informed of deviations from the expected state. Due to the current implementation this is prone to some erroneous warnings especially for older plates that have not been updated to the newest feature extraction procedures (see footnote 9), upon which reporting is based.

4.2.3 Creation of `Data` Structures

Irrespective of whether a given plate is already in cache or is requested for the first time, every `PlateData` structure is built from a `MatData` object. As mentioned in section 4.1.3, the motivation behind this is creating independence between the implementation of `PlateData` structures and the data itself and consequently no longer invalidating cached plate data by making changes to its representation. This requires the conversion of `MatData` into `PlateData` to be reasonably fast and much effort was invested in reducing the amount of time needed for this step.

The critical section in this process is creating the individual `ImageData` objects from the list structure resembling how single cell features are stored by CellProfiler. Listing 4.5 displays a segment of the code involved and while it might appear cluttered at first sight and the context of many variables is unclear, it is reproduced in order to highlight some important aspects. At this point in execution, among other things, the features have been sorted into groups according to their dimensions. Every feature is represented by a list containing as many slots as there are images on a plate and each slot holds a further list structure, the length of which is used for sorting. Single element objects are grouped into `vec` and handled by the code in lines 4–10, lists with multiple scalar elements are added to `mat` and treated by lines 12–29, while entries that constitute of a further nesting level are stored in `lst` and dealt with by lines 31–73. With this structure established, the outermost `apply`-type function sets up an index that moves through all lists simultaneously and in each step, the data corresponding to the given image is extracted and molded into an `ImageData` structure.

Scalar-valued features with respect to images are saved into a `data.frame`, as they typically present a mixture of numerical and character data. In order to speed up assembly, several checks are omitted that are performed by default when instantiating a `data.frame`. The predominant type of feature is vector valued at image level and therefore presents the most attractive target for optimization. Originally, here too, `data.frames` were created but switching to plain matrices, improves timings considerably. Special care has to be taken to ensure the correct dimensions of the resulting objects, preventing instances with only one member from being turned into vectors, as well as providing matrices with zero rows and a named column dimension in cases where no object is present. This level of consistency is important to reduce the amount of downstream code dedicated to checking dimensions and handling special cases.

The loop over list slot `mat` in line 14 iterates subgroups which are determined in order to create separate matrices for image objects with differing numbers

Listing 4.5: Due to the heterogeneity of InfectX datasets, build up of `PlateData` structure is fairly involved, as many edge cases have to be handled. Moreover, every instantiation of a `PlateData` object is preceded by the conversion from a `MatData` structure, requiring quick turnover. This code excerpt shows the section responsible for building `ImageData` objects from a nested list structure akin to what is produced by CellProfiler, by moving through all lists simultaneously and in building an image data object in every iteration. The three sections `vec`, `mat` and `lst` correspond to different types of features that are grouped according to their dimensions.

```
data$data <- plyr::llply(1:tot.nimgs, function(ind, data, name,
                                                n.imgs, quants, counts) {
  # image level data features
  if(length(data$vec) > 0) {
    vec <- data.frame(lapply(data$vec, function(feature, i) {
      return(feature[[i]])
    }, ind), stringsAsFactors=FALSE, check.names=FALSE, check.rows=FALSE)
  } else {
    vec <- NULL
  }
  # scalar valued single cell level data
  if(length(data$mat) > 0) {
    # subgroups corresponding to image objects with differing counts
    mat <- lapply(data$mat, function(group, ind) {
      n.rows <- length(group[[1]][[ind]])
      grp <- vapply(group, function(feature, i) {
        return(unlist(feature[i]))
      }, double(n.rows), ind)
      names <- colnames(grp)
      if(is.null(names)) names <- names(grp)
      n.cols <- length(names)
      dim(grp) <- c(n.rows, n.cols)
      colnames(grp) <- names
      rownames(grp) <- NULL
      return(grp)
    }, ind)
  } else {
    mat <- NULL
  }
  # vector valued single cell data
  if(length(data$lst) > 0) {
    lst <- mapply(function(group, gname, ind) {
      if(gname == "IdentityOfNeighbors") {
        return(lapply(group, function(feature, i) {
          # build sparse adjacency matrices
          l <- length(feature[[i]])
          if(l > 1) {
            p <- c(0, cumsum(sapply(feature[[i]],
                                    function(x) length(x[[1]]))))
            j <- unlist(feature[[i]])
            return(sparseMatrix(j=j, p=p, dims=c(l, l)))
          } else return(NULL)
        }, ind))
```

```
44        } else if(gname == "PercentTouchingNeighbors") {
45          return(lapply(group, function(feature, i) {
46            # only return percentage values
47            if(length(feature[[i]]) > 1) return(unlist(feature[[i]]))
48            else return(NULL)
49          }, ind))
50        } else {
51          return(lapply(group, function(feature, i) return(feature[i]), ind))
52        }
53      }, data$lst, names(data$lst), list(ind=ind), SIMPLIFY=FALSE)
54    } else {
55      lst <- NULL
56    }
57    # match pairs of features for generating percentage adjacency matrices
58    if(!is.null(lst$PercentTouchingNeighbors) &
59       !is.null(lst$IdentityOfNeighbors)) {
60      lst$PercentTouchingNeighbors <- mapply(function(feat, fname, mats) {
61        object <- unlist(strsplit(fname,
62                                  "Neighbors.PercentTouchingNeighbors"))[2]
63        mat <- mats[[grep(paste0("Neighbors.IdentityOfNeighbors", object),
64                          names(mats))]]
65        if(is.null(mat)) {
66          return(feat)
67        } else {
68          return(sparseMatrix(j=mat@i, p=mat@p, x=feat, dims=dim(mat),
69                 index1=FALSE))
70        }
71      }, lst$PercentTouchingNeighbors, names(lst$PercentTouchingNeighbors),
72         list(mats=lst$IdentityOfNeighbors), SIMPLIFY = FALSE)
73    }
74    return(ImageData(name, ind, n.imgs, quants, counts[ind], vec, mat, lst))
75  }, data$data, getBarcode(plate), n.imgs, data$meta$counts.quantiles,
76  cell.counts, .progress=progress.bar)
```

per well. While there are typically as many cells as there are nuclei, the count of pathogen objects will be different and corresponding features therefore have to be saved into separate structures. Naming of these subgroups is determined by majority rule of involved object types. Furthermore, the use of `vapply` instead of `lapply` in line 16 yields a critical performance improvement by pre-specification of the return type which is a numeric vector of length `n.rows`.

Finally, features that consist of nested lists at cell level are handled either by building adjacency matrices in case of neighbor features or by preserving their nested structure and attaching them as-is. Identification of neighbor features, unfortunately, constitutes one of the rare occurrences of resorting to hard-coded name-matching. IdentityOfNeighbors features are stored as a list with

one slot per cell, in turn containing a list of indices of all cells that have been identified as neighbors. The code in line 38 converts this information into row pointers and together with the vectorized form of all indices, this can be directly used to create a sparse matrix in compressed row storage format.

The second type of specially handled neighbor features, PercentTouchingNeighbors, cannot be addressed in the same loop, due to the incomplete list of lists (LIL) storage that was chosen for representation. Regular LIL format requires each value to be associated with the column index while the row index is provided by the encompassing list (or vice versa). In this case, the nested index is omitted, as it is already contained in the corresponding IdentityOfNeighbors feature. Therefore identity features are processed first and only when guaranteed to be available, a second loop matches pairs of neighbor features, reuses the sparsity structure and populates the new matrix with linearized percentage values (line 68). Iterating twice could be avoided, by sorting the features but as only few are available per plate this makes little difference in performance.

The resulting data in `vec`, `mat` and `lst` is used to build one `ImageData` object per iteration, which is returned to the parent environment. The other variables in line 74 constitute the image level metadata. This segment typically takes about a minute to execute, which is complemented by roughly another minute spent on reading a `MatData` cache file from disk. Assembling image objects into `WellData` structures is quick as only a regrouping of already built up objects is required. The major bottleneck here is fast generation of `WellMetadata` objects, one of which is attached to every `WellData` structure. This is achieved by using metadata caches in the form of `PlateAggregate` objects (see section 4.1.3). While there still is room for improvement, current performance is deemed sufficient for interactive usage and together with well-level caching, waiting times for data provisioning steps are minimal.

The reordering of data that is performed in the code segment shown requires roughly double the memory of a complete plate dataset, restricting usage of singleCellFeatures to hardware with upwards of 16 GB of RAM available. Some attempts were made at reducing the memory footprint which theoretically need not be as high. Whenever an `ImageData` object is instantiated, the corresponding raw data could be evicted from memory, cutting maximal memory usage back in half. Unfortunately such schemes so far have been met with significant performance impairments, leading to acceptance of memory limitations for the time being. Parallelization of `PlateData` object generation was explored but could not be implemented in a way that guaranteed stable performance. The main problem is that forked R processes have no way of triggering garbage collection among each other and it can happen that one process starves the others of available memory, slowing execution significantly.

There are many more issues that have to be handled when building `PlateData` objects from single cell features, due to considerable heterogeneity among InfectX datasets. One small example is that there is no way of knowing whether 6 or 9 images are available per well other than looking at the length of feature lists (2304 slots correspond to 6 images and 3456 slots are obtained with 9 images). Another factor, however can influence list length, as CellProfiler omits empty slots at the end. Taken together this means that the number of slots alone cannot be used to deduce the number of images per well with certainty. Of course, a simple heuristic resolves the issue, as it is very unlikely that the last > 1000 images on a plate are empty, but such aspects have to be dealt with nevertheless.[10]

4.3 Dataset Manipulation

Functionality surrounding manipulation of singleCellFeature data objects can be divided into 4 categories. First, there are augmentation and normalization functions used to generate new or modified data from existing features by aggregation or contextualization. Extraction and clean-up functions serve to distill the data and remove unneeded bulk, while validation procedures are implemented in order to check completeness and consistency of the data.

Lastly there is a function `meltData` which turns a `data` object into the smallest possible set of `data.frames` for making the data available to the rich environment of statistical analysis capabilities presented by R. A helper function accompanying data melting is also available and can be used to massage a molten structure into a single final `data.frame`, ready for external analysis. The following subsections discuss each of the outlined categories.

4.3.1 Feature Augmentation and Normalization

Location features by themselves are useless, as they are represented as two vectors of individual coordinate values. If these coordinates, however are somehow contextualized, they hold the key to a considerable amount of information. Two examples of coordinate augmentation functions are visualized in figure 4.4. The top view is created by separating data points into concentric ellipses which might be relevant for normalizing against technical issues, such as optical properties of microscope lenses that degrade towards image borders (e.g. vignetting and sharpness). The bottom diagram shows 2-dimensional kernel

[10] As a matter of personal opinion, I find the benefits of saving minute amounts of storage by occasionally omitting a handful of empty list slots at the end of some feature vectors to be clearly offset by the amount of code needed to handle resulting corner cases.

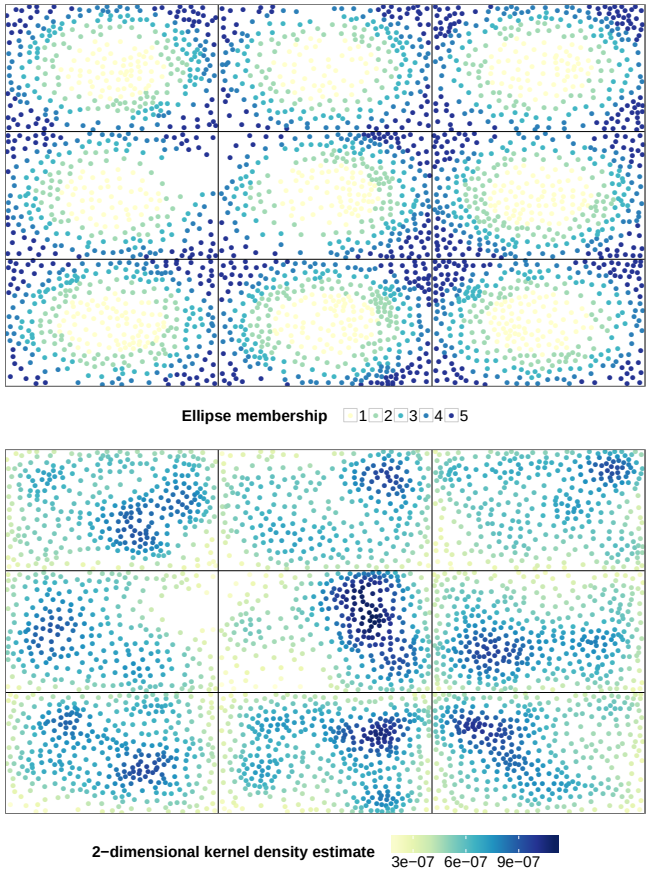


Figure 4.4: Two examples of coorinate augmentation functions aiming at producing useful information from coordinates of cellular objects. The first image shows the discretized elliptic distance (5 bins of equal area) from individual image centers which could be relevant because of deteriorating optical properties of microscope lenses, while the second diagram visualized the 2-dimensional density of cellular objects, an important population context feature. Data is taken from well H6 of plate J107-2C.

density estimates as calculated by the CRAN package sm (Bowman and Azzalini 1997) which represents another population context feature as proposed by Knapp et al. (2011) and Snijder et al. (2012).

Several additional features can be generated from object locations, all handled by `augmentCordinateFeatures`, including ellipses with respect to well center (instead of individual images centers), continuous distance from both image and well centers, as well as membership to 2-dimensional rectangular bins (both at well and image levels). Binning is accompanied by annotation with properties such as number of empty neighboring bins (see introductory example to this chapter) or their location within image and well (e.g. image border, image corner, image border that coincides with well border, image edge that is internal with respect to well, etc.).

The location of image within well can be used to generate a set of shifted coordinate features in order the describe object location not with respect to site, but well (calculated by `augmentImageLocation`). All these procedure have to be implemented in a very efficient manner, as they are used to process $\sim 10^6$ cells and cannot take more than seconds per feature in order to be used interactively. Some aspects of this have already been covered earlier and while there is much more to discuss, the interested reader is directed to the publicly hosted source code. All procedures can be applied at plate, well, and where applicable, at image level, owing to the S3 method dispatch system and the recursive implementation of data structures outlined in figure 4.2.

Another set of augmentation functions is directly aimed at data normalization. First there is `augmentAggregate`, which serves to calculate aggregate values on groups of objects (for example all cells in an image, all infected cells in a well or all pathogens throughout a plate). One argument that can be supplied as `func.aggr`, identifies an arbitrary function that produces a scalar from a vector of values (e.g. `min`, `max`, `mean`, `median`, `var`, `mad`, `sum`, etc.) and thusly performs the data summarization. An additional capability of note is the ability to include the 4 neighbors (north, east, south, west) when operating at well level in order to somewhat stabilize the results.

The function `augmentBscore` is only able to operate at plate level and produces row, column and plate effect estimates for each of the features specified by applying the `medpolish` procedure of the R stats package. Technical artifacts in HTS experiments can manifest as horizontally (for example due to a clogged pipette) or vertically striped patterns (e.g. degradation of microscope illumination) and B-scoring is a proven approach for correcting such issues. As is customary in B-scoring, control wells were originally excluded from the procedure. Unfortunately the way in which the layout of many InfectX plates is designed, this causes problems when control wells are to be used for analy-

sis, as they occupy entire rows/columns for which the corresponding row and column effects are unavailable. Interleaving siRNA wells with control wells would alleviate the problem but place siRNA wells at the plate border, which is undesirable as well. Currently, control wells are included in B-scoring.

In order to perform data normalization itself, the user has two procedures at his disposal: `normalizeData`, which may be applied at every hierarchy level and `augmentMars`, which is only available to `PlateData` objects. The former function can accept regular expressions to specify the set of features it is to be applied to, as well as two sets of features that may have been generated by `augmentAggregate` for scaling and centering of the data. A small example might better convey what is possible with `normalizeData`:

```
# fetch a complete plate dataset
data <- PlateData(PlateLocation("J107-2C"))
# discard any non-infected cells
data <- extractCells(data, infected=TRUE)
# regular expressions for selecting AreaShape, Intensity and Texture
# features of cellular objects
feat.sel <- c(".AreaShape_", ".Intensity_", ".Texture_")
feat.drp <- c("^Bacteria.", "^BlobBacteria.")
# calculate plate level median values for all selected features
data <- augmentAggregate(data, features=feat.sel, drop=feat.drp,
                         func.aggr="median", level="plate")
# calculate well level mad values for all selected features
data <- augmentAggregate(data, features=feat.sel, drop=feat.drp,
                         func.aggr="mad", level="well", neighbors=TRUE)
# normalize all selected values by robust Z-scoring
data <- normalizeData(data, select=feat.sel, drop=feat.drp,
                      values=".",
                      center="Aggreg_P_median$",
                      scale="Aggreg_N_mad$")
```

Any one of the center/scale parameters may be set to `NULL` and together with the flexibility of `augmentAggregate` a great range of possible normalization schemes can be applied. In this example a robust variant of Z-scoring is used, but only centering or scaling data is readily possible, as is using other summary functions (for example, `mean` and `var`, which would lead to regular Z-scoring).

One noteworthy implementation aspect is concerned with the great number of regular expressions that have to be executed. For each image, the set of selected features is iterated and within each iteration, the corresponding center and scale feature has to be found among the union of all available features

(which, due to the non-destructive operation of all augmentation functions can grow up to 2000–3000). Furthermore, features are organized in a nested list (remember: at top level, they are sorted into `data.vec`, `data.mat` and `data.lst` categories and within each further subdivision into groups according to number of objects is possible). Constant traversal and matching of the same feature pairs (or triplets) is inefficient, as it is identical for all images. Therefore, a stencil of feature indices that specify locations within the nested structure, is precomputed and applied for each individual image. This optimization makes execution of a normalization scheme over a complete plate possible within 90 s for a setup as in the above example (only infected cells) or 2 min for a full dataset.

The second normalization procedure, `augmentMars`, fits a fixed multivariate adaptive regression splines (MARS) model for each selected feature and returns the residuals as normalized feature values. Model fitting is performed by the CRAN package earth (Hastie and Milborrow 2015) and due to the large number of individual models that are involved, this is the most expensive operation in any singleCellFeatures analysis routine, taking on the order of 25 min per plate. Normalization will be discussed more in depth in chapter 5.3.

All augmentation and normalization functions operate in a non-destructive manner in the sense that they leave the original values intact and save results to copies of the features they were specified to operate on. In the above example, all features processed by the first application of `augmentAggregate` are saved with the added suffix `Aggreg_P_median`, which is used by `normalizeData` for identification as centering value. It is left to the user to discard sets of features that are no longer needed. While some space savings are possible, owing to the nested hierarchical model of `PlateData` structures (for example row and column effects of B-scoring do not have to be duplicated for every object, as would be the case in an entirely matrix based setup), objects holding several thousand features can become unwieldy. This is usually not a problem, as only a subset of original and derived features are needed leading to elimination of intermediate values by the user.

4.3.2 Data Filtering

Several functions for filtering datasets are available. Extraction of individual cells is possible via `extractCells` which may be applied at every plate, well or image level. Currently, the only available criterion is infection, but addition of a further group of parameters such as a vector of features, a set of order relations and corresponding thresholds could be added easily and would allow for arbitrary filtering with cellular granularity. At the next hierarchical tier, `extractImages`, available to `PlateData` and `WellData` objects, can be used to

select individual images either keeping the superordinate data structure intact or discarding it, yielding a naked list. Finally, data extraction at well level is possible using `extractWells`, which owing to circumstance is only available to plate objects. Again the user may keep the encompassing plate structure or throw it out.

Dropping or extracting a subset of features may be accomplished by calling the function `extractFeatures` on any type of `Data` structure. Either a vector of regular expressions can be used to select features, followed by application of a further vector of regular expressions for removing erroneously selected instances, or a vector of feature names can be supplied that is matched exactly. Again, for performance reasons, when run at plate scale, the large number of repeated regular expressions causes speed issues and in such cases, the feature names are worked out once for the first and reused in all subsequent wells.

The function `cleanData`, which may act on plates and wells, is currently used for eliminating image sites with very few or too many cellular objects. This measure serves to combat segmentation issues and eliminate some technical artifacts such as bad focus. Images are selected according to image-level cell count quantiles and the user can choose whether to do away with only the upper or lower 5%, or discard data symmetrically. Having well quality annotation metadata available, the procedure is planned to be extended to also remove wells that carry the label `BAD` but as of yet this has not been a priority due to the predominance of missing labels.

Further, related functionality is provided by `makeFeatureCompatible`, which can be used to ensure that multiple `Data` objects, supplied as a list, share the same features. For dealing with data heterogeneity, the intersection of individual feature sets is computed using the base function `Reduce` and subsequently extracted from each object. In order to illustrate the brevity with which such procedures can be implemented owing to R's broad range of library functions, coupled with convenience functions available to singleCellFeatures, the source to this function is reproduced in listing 4.6. This procedure can be applied to any combination of `PlateData`, `WellData` and `ImageData`, owing to generic functions and the S3 method dispatch system.

4.3.3 Ensuring Dataset Consistency

Despite efforts to design data structure modifying functions in such a way that prevents introducing inconsistencies, there is no formal mechanism of safeguarding against unintended manipulations and due to how S3 objects are designed, the user can do whatever he pleases (for example removing features only in a subset of wells). Therefore, several procedures exist, that can be used

Listing 4.6: Used for reducing the supplied list of objects to the common set of features, this function serves as an illustration of the conciseness that is possible due to R's range of library functions coupled with functionality implemented in singleCellFeatures.

```
1  makeFeatureCompatible <- function(lst) {
2    # input validation
3    stopifnot(sapply(lst, function(x) any(class(x) == "Data")))
4    # check objects for internal consistency
5    stopifnot(sapply(lst, checkConsistency))
6    # get individual feature lists
7    features <- lapply(lst, getFeatureNames)
8    # produce intersection of feature lists
9    intersection <- Reduce(intersect, features)
10   # extract intersection from each object
11   res <- lapply(lst, extractFeatures, features=intersection)
12   return(res)
13 }
```

to make sure that some assumptions made throughout the code base (such as an identical set of features throughout a plate) still hold and that the datastructures are valid. Checking whether the set of features per se is complete can be done with `checkCompletenessFeature`, which has been introduced previously as the function that analyzes `PlateData` objects before well caches are written (or prevents this from happening if problems are detected) and reports on the state of `MatData` caches following their retrieval.

Both functions `checkCompletenessImage` and `checkCompletenessWell` are concerned with structural integrity of the respective structures in the sense that they contain the expected number of sub-structures. In case of images, two possibilities exist, which have to be handled, whereas plates are currently fixed to a 384 well-layout. Finally, `checkConsistency` is slightly more involved, as no fixed template exists that can be matched. At well level, the method builds an image-level prototype consisting of metadata such as well name, plate barcode and the structuring of features alongside feature names themselves and compares this among all images. At plate level, the procedure is applied recursively, and requires 384 additional comparisons to check coherence among wells.

Listing 4.6 contains an example application of a consistency check (*cf.* line 5), which is required due to reliance on `getFeatureNames`. As this function is employed in several speed-critical settings, it simply extracts the feature set of a single image and does not check if every other image contains the same set of features. Due to the intent behind `makeFeatureCompatible` this however has to be enforced.

4.3.4 Data Melting[11]

One of the most important functions for every analysis procedure that relies on external tools, is `meltData` which turns the supplied `Data` structure into the smallest possible set of `data.frames` (see listing 4.7). At image level, the data representation does not have to be significantly altered apart from merging of some metadata fields into the data nodes in preparation of loss of structure conveying this information (examples include image and well indices, as well as plate barcode). At well level, the corresponding data structures are combined using `rbind` (for `vec` and `mat` features), while the `bdiag` function (from the Matrix package) is used for creating block diagonal adjacency matrices.

Well metadata is written into a special group under the `vec` node, thereby avoiding replication and consequent storage blowup (see listing 4.7, line 8 and following). A similar procedure for both storing metadata and concatenating subordinate data-structures is applied at plate level, yielding a nested list grouped by top level slots `vec`, `mat` and `lst`, which are further subdivided into `data.frames` corresponding to objects with different counts (Bacteria, BlobBacteria and Cells in case of the `mat` node shown in listing 4.7).

Acting on the structure resulting from running `meltData`, is the utility function `moveFeatures`, which is capable of relocating features between nodes `vec`, `mat` and `lst`. It can be used, for example, when a metadata feature, such as gene name needs to be available per cell, or the opposite way round, when a cellular feature should be included in the matrix representing per well features. Whether the selected data vectors have to be expanded, or collapsed, the function automatically determines which replication pattern to use or how to aggregate data (using the specified summary function) in order to meet the target dimensions. In practice, this is of great usefulness, combining space saving attributes of the molten data structure with the ability to expand user selected features for simplified interface with external procedures.

4.4 Utility and Convenience Functions

A multitude of specialized convenience functions for working with the described S3 objects were developed alongside to the functionality discussed above. Much of this code is used internally throughout the project, and where sensible, the methods are exported to the package namespace and hence made available to the user. Examples are functions for visualizing data, utilities for managing the package and cache objects, various getter functions to directly

[11] Naming of this functionality is inspired by the pair of functions `cast` and `melt`, provided by the CRAN package reshape2 (Wickham 2007).

Listing 4.7: While not a proper code listing, this represents a heavily truncated output view as produced by applying `str` on the result returned by `meltData` after having processed a complete plate.

```
List of 3
 $ vec:List of 3
  ..$ Image:'data.frame': 3456 obs. of  32 variables:
  .. ..$ Image.Count_Bacteria    : num [1:3456] 289 156 137 19 0 62 36 37 ...
  .. ..$ Image.Count_BlobBacteria: num [1:3456] 288 125 137 19 0 46 29 25 ...
  .. ..$ Well.Index              : num [1:3456] 1 1 1 1 1 1 1 1 2 ...
  .. .. [list output truncated]
  ..$ Well :'data.frame': 384 obs. of  15 variables:
  .. ..$ Well.Index     : num [1:384] 1 2 3 4 5 6 7 8 9 10 ...
  .. ..$ Well.Gene_ID   : chr [1:384] "523" "none" "none" "none" ...
  .. ..$ Well.siRNA_Name: chr [1:384] "ATP6V1A" "SCRAMBLED" "MOCK" "MOCK" ...
  .. .. [list output truncated]
  ..$ Plate:'data.frame': 1 obs. of  6 variables:
  .. ..$ Plate.Barcode  : chr "J101-2C"
  .. ..$ Plate.Quality  : chr "UNKNOWN"
  .. ..$ Experiment.Name: chr "BRUCELLA-DU-K1"
  .. .. [list output truncated]
 $ mat:List of 3
  ..$ Bacteria    :'data.frame':  388625 obs. of  51 variables:
  .. ..$ Bacteria.AreaShape_PerObjArea_CorrPathogen      : num [1:388625] ...
  .. ..$ Bacteria.Intensity_MassDisplacement_CorrPathogen: num [1:388625] ...
  .. ..$ Well.Index                                      : num [1:388625] ...
  .. .. [list output truncated]
  ..$ BlobBacteria:'data.frame':  222981 obs. of  7 variables:
  .. ..$ BlobBacteria.Location_Center_X: num [1:222981] 1.25 42.45 3 5 6.33 ...
  .. ..$ BlobBacteria.Location_Center_Y: num [1:222981] 130 210 119 111 138 ...
  .. ..$ Well.Index                    : num [1:222981] 1 1 1 1 1 1 1 1 1 1 ...
  .. .. [list output truncated]
  ..$ Cells       :'data.frame':  945851 obs. of  529 variables:
  .. ..$ Cells.AreaShape_Area        : num [1:945851] 141389 2104 2466 ...
  .. ..$ Cells.AreaShape_Eccentricity: num [1:945851] 0.893 0.554 0.928 ...
  .. ..$ Cells.AreaShape_EulerNumber : num [1:945851] 1 1 1 1 1 1 1 1 1 1 ...
  .. .. [list output truncated]
 $ lst:List of 2
  ..$ IdentityOfNeighbors     :List of 3
  .. ..$ Neighbors.IdentityOfNeighbors_Cells_2
                  :Formal class 'lgCMatrix' [package "Matrix"] with 6 slots
  .. .. .. .. ..@ i       : int [1:4746004] 8 4 7 2 6 9 19 4 9 10 ...
  .. .. .. .. ..@ p       : int [1:945843] 0 1 1 2 3 6 7 11 12 13 ...
  .. .. .. .. ..@ Dim     : int [1:2] 945842 945842
  .. .. .. .. ..@ Dimnames:List of 2
  .. .. .. .. .. ..$ : chr [1:945842] "A1_2" "A1_2" "A1_2" "A1_2" ...
  .. .. .. .. .. ..$ : NULL
  .. .. .. .. ..@ x       : logi [1:4746004] TRUE TRUE TRUE TRUE TRUE TRUE ...
  .. .. .. .. ..@ factors : list()
  .. .. [list output truncated]
  .. [list output truncated]
```

access and extract certain information from custom classes and conversion procedures for certain object pairs. Due to diversity and extent these additional features unfortunately cannot be covered in their entirety and only a selection is highlighted and briefly discussed.

Visualization. Heatmaps and box-plots can be used to visualize certain features in a plate wide context and bubble plots at a single image level are currently under development. An exemplary view, resulting from applying the function `plateHeatmap` to the feature Nuclei.AreaShape_Area, aggregating data at well level by calculating the mean and using a logarithmic color scale is shown in figure 4.5. Control wells are marked by black borders, while actual siRNA experiments are indicated by white rectangles. Tile coloring can be specified by supplying the function with a vector of colors that is interpolated to produce a gradient and the color scale is chosen by a function valued parameter (e.g. `identity` which is default, `log` or `sqrt`), while the data summary method is indicated analogously (e.g. `mean`, `median`, `var`, `mad`, etc.).

Plate-level box plots are a bit unwieldy for print media, are therefore not shown, but provide a useful method of spotting wells which might contain untrustworthy data. Bubble plots, thought for overlaying individual images with some feature data are partially implemented but the automatic image fetching from openBIS still requires some work. Such visualizations are primarily used for validating sensibility of feature data, by providing side-by side comparison with raw image data.

Package utilities. Functions of this group have been mentioned in several places and comprise of tools for metadata coverage assessment, configuration management, cache updating and invalidation, as well as database management. A report on the discrepancy between plates that have single cell features available and datasets that are represented by metadata is generated each time the package is attached to an R session (using the `.onLoad` hook). The corresponding function, run as `wellDatabaseCoverage(TRUE)`, can be used to display a detailed overview of current metadata coverage at well level which is particularly important for searching.

Configuration management relies on a yaml based file used storing system specific parameters (see section B.1). Setter functions, as well as getters for location and content, in addition to an initiation routine that sets up an empty template for user editing, are available for its manipulation and access.

In order to curate the lookup tables used for search and various speed critical, metadata dependent processes, each database type has a corresponding update function available (`updateDatabaseFeatures`, `updateDatabasePlate` and

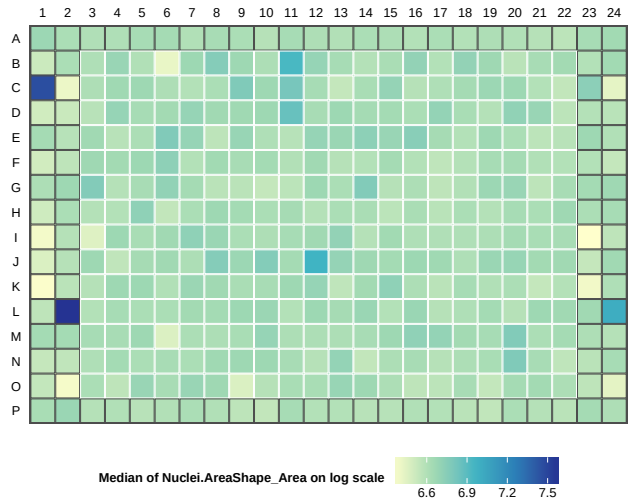


Figure 4.5: Several visualization procedures are integrated with singleCellFeatures. In this example, the median of the feature `Nuclei.AreaShape_Area` is shown on a logarithmic scale. Two parameters of `plateHeatmap` are functions so that the user can customize both, how data per well is summarized and on what scale colors are determined. The plate is J101-2C (wells C1, L2, C23 and L24 contain cell killer Kif11 siRNA).

`updateDatabaseWells`). As all metadata available to singleCellFeatures is static and does not reflect new additions to openBIS, section B.3 describes how to update the underlying files and the above tools can recreate the databases using updated source data.

While cache-related functionality has primarily been covered from a perspective of creation and retrieval, several additional functions for managing the set of cached objects are at the users disposal. Metadata and well caches can be flushed (rebuilding is comparably cheap) and reports on the extent of plate level caches can be compiled. Moreover, `rebuildAllMatDataCaches` implements a procedure that iterates all plate cache objects and determines if any improvements are available. This can be used, for example after batch-downloading many plates to trigger retry of features that failed to import in applicable plates, or in case some features did not download, selectively initiate re-fetching. A further application is if new features become available, the function can be used to specifically start the respective downloads and imports into existing objects eliminating the need of having to re-process affected plates completely (which is an expensive operation).

Listing 4.8: Many convenience functions are available to singleCellFeatures, one of which can be used to convert its argument into a `PlateLocation` object. A further example of such a simple, single purpose method is the getter function `getBarcode`, which extract the barcode from the object it is called upon.

```
convertToPlateLocation <- function(x) {
  UseMethod("convertToPlateLocation", x)
}

convertToPlateLocation.DataLocation <- function(x) {
  barcode <- getBarcode(x)
  return(PlateLocation(barcode))
}

convertToPlateLocation.Data <- function(x) {
  barcode <- getBarcode(x)
  return(PlateLocation(barcode))
}

convertToPlateLocation.PlateAggregate <- function(x) {
  return(x$plate)
}

convertToPlateLocation.default <- function(x) {
  stop("can only deal with PlateData/MatData/WellLocation objects.")
}
```

Furthermore, for the package to be usable in a cluster environment, resource allocation has to be respected. Parallelized sections are using foreach (Weston and Calaway 2014) via the doParallel backend (Weston, Calaway, and Tenenbaum 2014) and it needs to be ensured that forked processes only run on the allowed cores. The function `getNumCores`, for example, can be used to determine the number of available cores by either reading the corresponding environmental variable set by Platform LSF, or by using `detectCores` of doParallel.

Convenience functions. This section covers a diverse range of methods including many getters for S3 objects and short, narrow purpose functions for common tasks. It is unfitting to describe these tools in their entirety and the interested reader is encouraged to look at the publicly hosted source code, or look though the manual pages that come with the package and are browsable through the R help system. Listing 4.8 shows one instance of such a function which can be used to create a `PlateLocation` object out of the supplied data structure, thereby converting the argument into a plate location specification. The example on page 101 provides a real-world application of such a conversion function.

The reasoning for using multiple description attributes for objects can be illustrated by means of this code excerpt:[12] As the function `getBarcode` (one of the aforementioned getter functions) is defined for all `Data` classes, four functions acting on `MatData`, `PlateData`, `WellData` and `ImageData` can be written as one by creating this additional level of hierarchy in object type naming, thereby reducing redundant code. In this case, one could even combine the functions that act on `DataLocation` and `Data` objects if an attribute grouping those classes were available. However, to avoid overly complex attribute relationships, only functionally related classes are gathered.

Interface to GLM routines. Currently still under development, several functions that interface with GLM routines are available to the user. Originally intended for building a model that discriminates two gene knockdown experiments based on single cell features, the function `prepareDataforGlm` can take two lists of wells and make the data ready for analysis with GLM by concatenating and annotating the wells with a response vector. Additional options are sampling based splitting of data into test and training sets and dropping of features that by themselves separate the data into the two knockdown groups.

Due to issues in logistic regression, caused by data that is separated into the two groups that are modeled by a hyperplane, `analyzeSeparation` can be used for preliminary investigation of this aspect. The procedure will first check if individual features separate the data, followed by looking at all $\binom{n}{2}$ pairs of features. Unordered k-tuples may be explored up to a user defined threshold but with several 100 features to be considered, the number of sets resulting from $k > 2$ becomes prohibitively large and the function will skip iteration k and above if $\binom{n}{k} > 250000$. In a final step, all features are considered simultaneously, solving the problem with a linear programming approach, using the CRAN package lpSolveAPI (Konis 2007). A further data issue that has to be dealt with is rank deficiency and the procedure `makeRankFull` can be used in this capacity. Features that either have zero variance or comprise of highly correlated columns, are removed prior to analysis.

A parallelized function for investigating the stability of the largest n coefficients in GLM, the set of features as determined by stepwise GLM, or the set of nonzero coefficients in regularized GLM analysis is available as the function `glmBootstrapStability`. For both stepwise and plain GLM, the functions `glm` and `step` of the R stats package are used, while regularized fitting is performed by `glmnet` (Friedman, Hastie, and Tibshirani 2010). A user specified fraction of the data (default 0.7) is sampled in each iteration and the resulting coefficients

[12] As a reminder, for instance, both `PlateLocation` and `WellLocation` are also `DataLocation` objects and the same pattern holds for other groups of S3 classes.

are stored for subsequent counting and tabulation. Unfortunately, there currently are unresolved issues regarding data normalization, making this type of analysis infeasible. Therefore all GLM oriented routines are subject to change.

Chapter 5

Data Analysis

The initial goal of this master's thesis was concerned with modeling different infection patterns as influenced through siRNA mediated gene knockdown. By fitting a GLM model, using the large set of available cellular features, described in section 3.4, as predictor variables and the membership to one of two wells as binary response, it was conjectured that it is possible to determine a subset of most influential features. Unfortunately, to present date, this could not be executed in a sensible manner.

In order the make the substantial amount of single cell data, obtained from several large-scale, high throughput siRNA screens performed by the InfectX consortium, available to an environment for statistical analysis, an R packages was developed, which is described in chapter 4. Building on this crucial piece of infrastructure, the current chapter will explore some of the data analysis that was performed, beginning a short introduction into the theory of GLMs and continuing with preliminary findings that motivate the investigation of desirable normalization schemes. A final outlook on potential improvements will conclude this chapter.

5.1 Statistical Models

Many algorithms for binary classification exist, including decision trees, SVMs, Bayesian networks, neural networks and GLMs, some of which have been encountered in previous sections due to their application in infection scoring (see section 3.5). As neither prediction not classification per se are of main interest, binary logistic regression presents an attractive method due to availability of closed-form coefficient and model statistics. Therefore, modeling of siRNA ef-

fects on cellular features is performed, using the glm function provided by the R stats package (R Core Team 2015).

A large number of binary comparisons are possible with the given datasets. In order to focus on attractive wells in the sense that there is reason to assume they might be biologically interesting, the $\binom{n}{2}$ possible combinations of wells (within a single plate, where $n = 384$, ~70000 pairs can be formed), are narrowed down using a PMM as derived by Rämö et al. (2014). Of the resulting hit list, several genes are selected and compared to wells containing scrambled siRNA reagents, which should provide a good choice for usage as control. A possible alternative to scrambled experiments are mock wells but owing to the complete absence of siRNA molecules, the difference in treatment of cells is only increased and hence it can be argued that they provide biased comparisons.

5.1.1 Generalized Linear Models

Modeling the relationship among variables is one of the most important applications of statistical theory. The study of regression analysis (and the closely related notion of correlation) started to form towards the end of the 19th century with Sir Francis Galton's study of height heredity in humans and his observation of regression towards the mean. Over the next few years, Udny Yule and Karl Pearson cast the developed concepts into precise mathematical formulation, in turn building on work performed by Adrien-Marie Legendre and Carl Friedrich Gauss who developed the method of least squares almost a century earlier (Allen 1997).

A multiple linear regression model can be written in matrix-vector form as

$$y = X\beta + \varepsilon \tag{5.1}$$

where $y \in \mathbb{R}^n$ is the vector of observations on the dependent variable, the design matrix $X \in \mathbb{R}^{n \times p}$ contains data on the independent variables, $\beta \in \mathbb{R}^p$ is the p-dimensional parameter vector and the error term $\varepsilon \in \mathbb{R}^n$ captures effects not modeled by the regressors.

In order to estimate unknown coefficients β_i, the ordinary least squares estimator minimizes the residual sum of squares, the squared differences between observed responses and their predictions according to the linear model.

$$\widehat{\beta} = \arg\min_{\beta} \|y - X\beta\|^2 \tag{5.2a}$$

$$= (X^T X)^{-1} X^T y \tag{5.2b}$$

Some assumptions are typically associated with linear regression models that yield desirable attributes for the estimates (e.g. statistical tests of significance). None of these restrictions are imposed on the explanatory variables; they can be continuous or discrete and combined as well as transformed arbitrarily. Furthermore, in practice, it is irrelevant whether the covariates are treated as random variables or as deterministic constants. With exception of the field of econometrics it appears that the majority of literature adheres to the latter interpretation and therefore, statements will not explicitly be conditional on covariate values.

Linearity. The relationship between dependent and independent variables is assumed to be linear in coefficients (after suitable transformations of regressors) and individual effects additive. If this cannot be satisfied, a linear model is not suitable.

Full rank. For the matrix $X^T X$ to be invertible, it has to have full column rank p. Therefore $n \geq p$ and all covariates must be linearly independent.

Exogeneity. All independent variables should be known exactly i.e. contain no measurement or observation errors as only the mean squared error of the dependent variable is minimized. Additionally, all important causal factors have to be included in the model. Exogeneity implies $\mathrm{E}[\varepsilon_i] = 0 \ \forall\, i$, as well no correlation between regressors and error terms (Hayashi 2000).

Spherical errors. This includes both homoscedasticity or constant error variance, $\mathrm{E}[\varepsilon_i^2] = \sigma^2 \ \forall\, i$, and uncorrelated errors $\mathrm{E}[\varepsilon_i \varepsilon_j] = 0 \ \forall\, i \neq j$. These two conditions can be written more compactly as $\mathrm{Var}(\varepsilon) = \sigma^2 I_{n \times n}$.

Normality. For the estimated coefficients to gain some additional desirable characteristics, it can be required that the errors ε_i be jointly normally distributed. Together with the above restrictions on expectation and variance, this yields $\varepsilon \sim \mathcal{N}_n(0, \sigma^2 I_{n \times n})$.

Violations of these assumptions have varying consequences. In situations of perfect collinearity, the ordinary least squares estimator $\widehat{\beta}$ as defined in (5.2b) does not exist. Recovering such a situation is possible by using a generalized matrix inverse (for example the Moore–Penrose pseudoinverse) or by dropping the corresponding variables. High correlation inflates coefficient variances, which may be countered by employing a regularization scheme like ridge regression. Failure to fulfill exogeneity, for example by omitting relevant explanatory variables, leads to regressors that are correlated with the error term, which in turn yields invalid (biased and inconsistent) ordinary least squares (OLS) estimates. The method of instrumental variables can help to produce an unbiased estimator nonetheless.

The assumption of spherical errors ensures that the least squares estimator is the best linear unbiased estimator in the sense that it has minimal variance among all linear unbiased estimators. Heteroscedasticity and autocorrelation do not yield biased coefficient estimates but can introduce bias in OLS estimates of variance, causing inaccurate standard errors. Such a situation calls for generalized least squares estimation, such as weighted least squares for when the data is not homoscedastic (σ^2 is vector-valued but off-diagonal elements of $\mathrm{Var}(\varepsilon)$ are zero), or feasible generalized least squares which can be applied in case of heteroscedasticity and/or correlation between errors. Whenever the condition of spherical errors does not hold, OLS is inefficient and generalized least squares estimators have smaller variance.

Finally, normality provides the framework necessary for applying common hypothesis testing, yielding t-statistics, p-values and confidence intervals for coefficients, as well as an F-statistic for the model as a whole. Furthermore, under normality assumptions, the OLS estimator and maximum likelihood estimator (MLE) coincide. Quantile regression and other forms of robust regression, such as M estimators may restore validity of inference when errors do not follow a normal distribution.

Modeling data to a dichotomous response variable with OLS methodology, while readily possible, often is a bad choice due to violations of several of the above assumptions. First of all, normal distributions are continous with support $x \in \mathbb{R}$ and thusly a categorical variable cannot be normally distributed. Homoscedasticity does not hold either, as can easily be seen in a geometric argument: any line with nonzero slope, fitted to a set of Y with $Y_i \in \{0, 1\}$, will produce residuals that vary linearly. Finally and perhaps most importantly, a linear model is unsuitable due to missing constraints on the range of fitted values. While this by itself could be dealt with, the concept of additivity within this context raises questions of its own when fitted values are thought of as probabilities. Linear behavior throughout the range of 0 (impossible) to 1 (certain) makes little sense for most practical applications, as typically, some flattening is expected when approaching either end of the spectrum.

In view of the above considerations it becomes clear that some sort of extension to ordinary linear regression is needed in order to deal with binary response variables. A theory, unifying several previously separately treated statistical models, including linear regression, analysis of variance (ANOVA), logistic regression, Poisson regression and multinomial response, was developed by John Nelder, Robert Wedderburn and Peter McCullagh in the early 1970's (Nelder and Wedderburn 1972; McCullagh and Nelder 1989) and is known as generalized linear model (GLM). Much of this section is based on (McCullagh and Nelder 1989).

Table 5.1: Common univariate distributions of the exponential family alongside mean and canonical link functions.

Distribution	Support	Link name	Link function	Mean function
Normal	$(-\infty, +\infty)$	identity	$\eta = \mu$	$\mu = X\beta$
Poisson	$\{0, 1, 2, \dots\}$	log	$\eta = \log(\mu)$	$\mu = e^{X\beta}$
Binomial	$\{0, 1, \dots, N\}$	logit	$\eta = \log\left(\frac{\mu}{1-\mu}\right)$	$\mu = \frac{e^{X\beta}}{1+e^{X\beta}}$
Gamma	$(0, +\infty)$	reciprocal	$\eta = -\mu^{-1}$	$\mu = -(X\beta)^{-1}$
Inverse Gaussian	$(0, +\infty)$	inverse squared	$\eta = -\mu^{-2}$	$\mu = (-2X\beta)^{-1/2}$

In order to accommodate the newly added extensions, the classical linear model is rewritten as

$$\mathrm{E}[Y] = \mu \text{ where } g(\mu) = \eta \text{ and } \eta = X\beta, \tag{5.3}$$

yielding a three-part specification, consisting of

(a) the *random component*; Y is distributed according to a member of the exponential family,[1]

(b) the *systematic component*; a linear predictor is given by $\eta = X\beta$,

(c) and the *link* between random and systematic components, expressed as the link function $g(\cdot)$, such that $\eta = g(\mu)$; in case of normally distributed Y, $\mu = \eta$ (identity link).

Mean and canonical link functions for several common univariate distributions belonging to the exponential family are shown in table 5.1. Binary response can

[1] Many of the predominately used distributions belong to the exponential family, including but not restricted to Bernoulli, binomial, Poisson, exponential and normal distributions. Common to all members, the probability density function can be written as

$$f(x;\theta) = h(x)e^{\theta^{\mathsf{T}} T(x) - A(\theta)} \tag{5.4}$$

where θ is the vector of parameters, $T(x)$ the vector of sufficient statistics, $A(\theta)$ the cumulant generating function and $h(x)$ the base measure. In case of the Bernoulli distribution

$$f(k;\pi) = \pi^k(1-\pi)^{1-k} \tag{5.5}$$

this gives $h(k) = 1$ $T(k) = k$, $\theta = \log\frac{\pi}{1-\pi}$ and $A(\theta) = \log(1+e^{\theta})$. Restricting GLMs to exponential family distributions makes it possible to stay within the framework of maximum likelihood parameter estimations, as given this restriction, MLE yields the best parameter estimator with respect to minimal variance.

be interpreted as

$$Y_i \sim \mathcal{B}\text{ernoulli}(\pi_i) \tag{5.6}$$

with

$$\pi_i = \mathcal{P}(Y_i = 1) = 1 - \mathcal{P}(Y_i = 0). \tag{5.7}$$

Therefore, π_i can be thought of as the probability of one outcome (i.e. success) and its complement $(1 - \pi_i)$ corresponds to the probability of the other outcome (failure). Each experimental unit that yields one outcome is associated with a vector of independent variables $x_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,p})$ and the goal of regression in this context becomes modeling the relationship between response probability π_i and explanatory variables x_i. A suitable link function for Bernoulli distributed response, as discussed above, is required to map the unit interval onto the whole real line and preferably be shaped sigmoidally in order accurately describe increasingly likely as well as increasingly unlikely events. Three possibilities are often considered within this context,

(a) the *logit* or logistic function

$$g(\pi) = \log\left(\frac{\pi}{1-\pi}\right), \tag{5.8}$$

(b) the *probit* or inverse normal function

$$g(\pi) = \Phi^{-1}(\pi), \tag{5.9}$$

with $\Phi(\cdot)$ denoting the cumulative distribution function of the normal distribution,

(c) and the *complementary log-log* function

$$g(\pi) = \log\left(-\log(1-\pi)\right). \tag{5.10}$$

All three functions are continuous and increasing on $(0, 1)$ and the first two are symmetric in the sense that $g(\pi) = -g(1-\pi)$, while the third is not. Using a logit link function, the probability of a positive response can therefore be written as

$$\pi_i = \frac{e^{x_i^\intercal \beta}}{1 + e^{x_i^\intercal \beta}} \tag{5.11}$$

and the model is, slightly rearranged, stated as

$$\log\left(\frac{\pi_i}{1-\pi_i}\right) = x_i^\intercal \beta = \sum_{j=1}^{p} x_{i,j}\beta_j. \tag{5.12}$$

where x_i is shorthand for the i-th row $(x_{i,1}, x_{i,2}, \ldots, x_{i,p})$ of the design matrix X. Consequently, of a unit change in a covariate $x_{i,j}$ will increase the corresponding probability log-odds by a multiplicative factor $\exp(\beta_j)$, as can easily be seen by exponentiating expression 5.12.

The likelihood of a set of parameter values π, given the data y, is equal to the probability of the data, given the parameters. Hence, in a logistic regression model (see expression 5.5 for the probability mass function of a Bernoulli distribution), we have

$$L(\beta; y) = \mathcal{P}(y \mid \beta) = \prod_{i=1}^{n} \pi_i^{y_i} (1 - \pi_i)^{1-y_i} \tag{5.13a}$$

$$= \prod_{i=1}^{n} \frac{e^{x_i^\intercal \beta y_i}}{1 + e^{x_i^\intercal \beta}} \tag{5.13b}$$

and expressed, for reasons of convenience, as log-likelihood

$$l(\beta; y) = \sum_{i=1}^{n} y_i \log(\pi_i) + (1 - y_i) \log(1 - \pi_i) \tag{5.14a}$$

$$= \sum_{i=1}^{n} x_i^\intercal \beta y_i - \log\left(1 + e^{x_i^\intercal \beta}\right). \tag{5.14b}$$

Both expressions 5.13b and 5.14b are derived using the logit link function but any other of the proposed links can be used to substitute π_i in equations 5.13a and 5.14a. In order to find the maximum likelihood estimates

$$\widehat{\beta} = \arg\max_{\beta} l(\beta; y), \tag{5.15}$$

the first derivative of $l(\beta; y)$ with respect to β_j in required:

$$\frac{\partial l(\beta)}{\partial \beta_j} = \sum_{i=1}^{n} \frac{y_i - \pi_i}{\pi_i (1 - \pi_i)} \frac{d\pi_i}{d\eta_i} \frac{\partial \eta_i}{\partial \beta_j} \tag{5.16a}$$

$$= \sum_{i=1}^{n} \left(y_i - \frac{e^{x_i^\intercal \beta}}{1 + e^{x_i^\intercal \beta}} \right) x_{i,j}. \tag{5.16b}$$

Again, 5.16b is obtained from 5.16a by using a logit link, and consequently substituting

$$\frac{d\pi_i}{d\eta_i} = \frac{-e^{-\eta_i}}{(1+e^{-\eta_i})},\ \eta_i = x_i^\intercal \beta \ \text{ and } \ \frac{\partial \eta_i}{\partial \beta_j} = x_{i,j}.$$

The log-likelihood maximum is found by setting the first derivatives in 5.16 to zero and solving for β. No closed form solutions to the resulting equations exist and therefore numerical algorithms such as Newton-Raphson are usually used, which under the given circumstances can be formulated as an iteratively reweighted least squares (IRLS) regression problem. Newton's method attempts to find the root x_r of a differentiable function $f(x)$ by starting with an initial guess x_0 and iterating

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}. \tag{5.17}$$

When a good initial guess is made and some restrictions on $f(x)$ apply, the rate of convergence is quadratic[2] and even if certain conditions do not hold, or the starting point is not suitably chosen, the algorithm may still converge towards the root. Returning to maximum likelihood notation of expression 5.16, a Newton-Raphson iteration can be written as

$$\beta^{(n+1)} = \beta^{(n)} - \frac{l'(\beta^{(n)})}{l''(\beta^{(n)})}. \tag{5.18}$$

The second derivative of the log likelihood with respect to β, as needed in 5.18 is

$$\frac{\partial^2 l(\beta)}{\partial \beta_j \partial \beta_k} = -\sum_{i=1}^{n} \frac{1}{\pi_i(1-\pi_i)} \left(\frac{d\pi_i}{d\eta_i}\right)^2 \frac{\partial \eta_i}{\partial \beta_j} \frac{\partial \eta_i}{\partial \beta_k}, \tag{5.19}$$

which, in matrix form, can be written as

$$\frac{\partial^2 l(\beta)}{\partial \beta \partial \beta^\intercal} = -X^\intercal W X, \tag{5.20}$$

[2]Given a function$f(x)$, that is three-times differentiable in an interval $I^* = [a, b]$ such that $a < x_r < b$ and $f'(x_r) \neq 0$, there exists an interval $I = [x_r - \delta, x_r + \delta]$, with $\delta > 0$ for which every $x_0 \in I$ converges quadratically towards x_r (Schwarz and Köckler 2006).

with the $n \times n$ diagonal matrix W

$$W = \text{diag}\left\{ \frac{1}{\pi_i(1-\pi_i)} \left(\frac{d\pi_i}{d\eta_i} \right)^2 \right\}, \text{ as } \frac{\partial \eta_i}{\partial \beta_j} = x_{i,j} \text{ and } \frac{\partial \eta_i}{\partial \beta_k} = x_{i,k}.$$

In the case of logistic regression, W simplifies to $w = \text{diag}\{\pi_i(1-\pi_i)\}$ and $l'(\beta)$ can be written in matrix form as $l'(\beta) = X^\intercal(y - \pi)$.

Using these results, expression 5.18 is rewritten as

$$\beta^{(n+1)} = \beta^{(n)} + \frac{X^\intercal(y-\pi)}{X^\intercal W X}, \tag{5.21}$$

which is iterated until the updates become smaller than a threshold and convergence is said to have been reached. In order to reformulate Newton-Raphson as an IRLS problem, $\beta^{(n)}$ is substituted by

$$\beta^{(n)} = \frac{X^\intercal W X}{X^\intercal W X} \beta^{(n)},$$

yielding

$$\beta^{(n+1)} = (X^\intercal W X)^{-1} X^\intercal W \left(X\beta^{(n)} + W^{-1}(y-\pi) \right) \tag{5.22a}$$

$$= (X^\intercal W X)^{-1} X^\intercal W z \tag{5.22b}$$

where expression 5.22b corresponds to a weighted least squares problem with response

$$z = X\beta^{(n)} + W^{-1}(y-\pi).$$

Therefore each iteration of equation 5.22b solves weighted regression on a transformed version of the response, called the adjusted dependent variable. At each step, the current estimate of β, $\beta^{(n)}$, is used to compute new weights W and consequently a new value for the adjusted variable z, which in turn is used for computing the minimizer for

$$\beta^{(n+1)} = \arg\min_{\beta} \left(z - X\beta^{(n)} \right)^{-1} W \left(z - X\beta^{(n)} \right), \tag{5.23}$$

providing new values $\beta^{(n+1)}$.

5.1.2 Parallel Mixed Model

As a way of exploiting multi-tier replication, typically involved in siRNA screening, starting at the level of individual sequences, different sequences targeting the same gene and as is the case for InfectX, different treatments in the form of varying pathogens, Rämö et al. developed a parallel mixed model (PMM), capable of gaining statistical power from structural parallelism. All following remarks are adapted from the referenced publication and the topic is only outlined briefly for completeness sake. The interested reader is directed to Rämö et al. (2014) for more information. The model as proposed is written as

$$y_{pgs} = \mu_p + a_g + b_{pg} + \varepsilon_{pgs}, \tag{5.24}$$

where y_{pgs} represents the per-well phenotype (e.g. infection score), which is described as the sum of a fixed effect μ_p for pathogen p, as well as random effects a_g for gene g, b_{pg}, a correction for gene g within pathogen p, and an error term e_{pgs}. The overall effect of gene knockdown g under pathogen treatment p therefore is

$$c_{pg} = a_g + b_{pg}, \tag{5.25}$$

and a positive estimated effect corresponds to enhanced infection levels, while a negative c_{pg} value indicates reduced infectivity. Random effects are distributed as $a_g \sim \mathcal{N}(0, \sigma_a^2)$, $b_{pg} \sim \mathcal{N}(0, \sigma_b^2)$ and $\varepsilon_{pgs} \sim \mathcal{N}(0, \sigma_\varepsilon^2)$, while estimation is carried out by the CRAN package lme4, using restricted maximum likelihood. Hits are selected according to an estimated local false discovery rate (FDR) which assumes that a mixture of two distributions corresponding to genes that are no hits (f_0) and genes that actually are hits (f_1), generates the overall distribution. Furthermore, it is surmised that $f_0 \sim \mathcal{N}(0, \sigma_a^2 + \sigma_b^2)$ and $f_0 \sim \mathcal{N}(\theta, \sigma_a^2 + \sigma_b^2)$, i.e. the two distributions are identical except for a shift in mean by θ. The overall distribution can be expressed as $f(c_{pg}) = \pi_0 f_0(c_{pg}) + (1 - \pi_0) f_1(c_{pg})$, where π_0 is the proportion of true hits. Finally, the FDR is defined as

$$\mathrm{fdr}(c_{pg}) = q_{pg} = \frac{\pi_0 f_0(c_{pg})}{f(c_{pg})} \tag{5.26}$$

and represents the probability that the effect for a given gene and pathogen is a false discovery. Figure 5.1 displays a visualization obtained by applying the PMM package (available on Bioconductor) to kinome-wide InfectX screens.

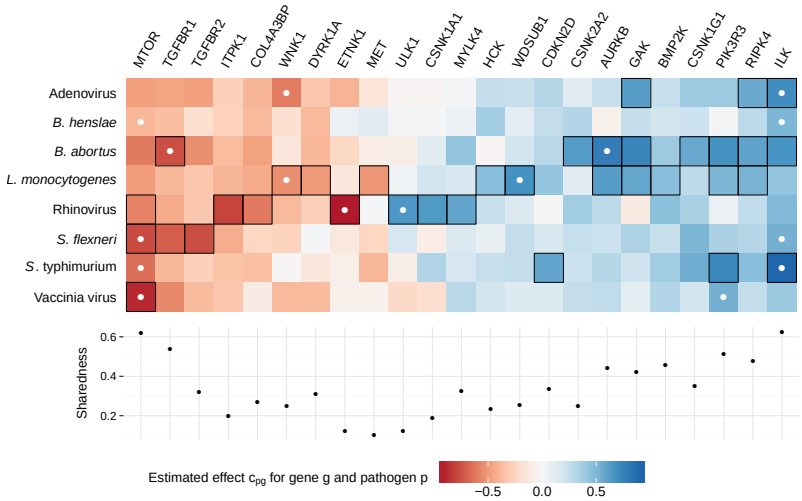


Figure 5.1: A heatmap plot as produced by the Bioconductor package PMM, which displays all genes that were determined to be significant hits (FDR < 0.4; indicated by black borders) for at least one pathogen. Genes are ordered by average c_{pg} values and both extrema are marked with white dots, while the sharedness score is shown as a scatterplot below. All available kinome screens were taken into consideration.

Color-coding corresponds to estimated c_{pg} effects and columns are sorted according to descending mean values. Only genes are included where the estimated FDR is below 0.4 for at least one pathogen (indicated by black borders), suggesting that up to 40% of individual hits may be superfluous. Centered white dots indicate the maximum and minimum c_{pg} value for each pathogen. For each gene, a sharedness score is displayed as well. This quantity is defined as

$$s_g = \frac{1}{2}\left(\left(1 - \frac{1}{P}\sum_{p=1}^{P} q_{pg}\right) + \frac{\sum_{p=1}^{P} \mathbb{1}_{q_{pg}<1}}{P}\right) \tag{5.27}$$

and quantifies how common a hit gene is among pathogens by describing both the extent of downward shift from 1 of the mean q_{pg} value (over all $P = 8$ pathogens), as well as the fraction of pathogens that contribute $q_{pg} < 1$ instances.

5.2 Preliminary Findings

Due to the high degree of redundancy in feature extraction during image analysis, a significant amount of correlation among features can be expected. Measurements of objects describing similar image segments, as well as features that build on related concepts, such as mean, median or integrated intensities, will obviously yield similar values and cause a dependence structure that has to be dealt with in statistical analysis. Figure 5.2 visualizes the issue by showing a heatmap representation of a correlation matrix, obtained from all available *AreaShape*, *Intensity* and *Texture* features for a *Brucella* plate using Dharmacon siRNA (J110-2D) and a randomly sampled subpopulation of cells (10%).

The three feature groups can easily be spotted as diagonal blocks with high within group and lower between group correlation, and the situation is worst for intensity features due to the extensive set of measurements quantifying similar properties (see figure 3.5). The texture segment is subdivided into four groups, corresponding from bottom left to top right to *Cells*, *Nuclei*, *PeriNuclei* and *VoronoiCells*. Absent off-diagonal correlation between nuclear and perinuclear regions is due to mutual exclusivity, whereas the off-diagonal correlation of cell and Voronoi cell features is due to significant overlap.

In order to deal with this data issue, features are transformed to the coordinate system of principal components (PCs) prior to GLM model fitting. When employing principal component analysis (PCA), typically the first 10% (or 30–50) of PCs capture around 90% of the overall variance in the data, corroborating the above claim of significant correlation among feature vectors. A further problem that is present in many datasets is that pairs of wells can be perfectly separated. This causes problems in maximum likelihood estimation, as affected coefficients are allowed to grow arbitrarily large, but is not unexpected given the large design space. PCA provides a tool for addressing the issue by encouraging only inclusion of a subset of PCs and thus reducing dimensionality.

Several GLM model fits, based on principal component regression are summarized in table 5.2. For the same plate is used in figure 5.2 (J110-2D, *Brucella*, Dharmacon unpooled, replicate 1), well pairs corresponding to the genes identified by PMM as down-hits, MTOR (H6) and TGFBR1 (M4), as well as up-hits, PIK3R3 (K8) and RIPK4 (G17), are formed with all available scrambled wells of the given plate. In order to establish a baseline of sorts, scrambled wells are also paired with each other.

For describing how well the discrepancy between the two input wells is captured, some model characteristics, alongside scores for predictions, are reported. The AIC is a goodness of fit estimate, constituting of the maximized

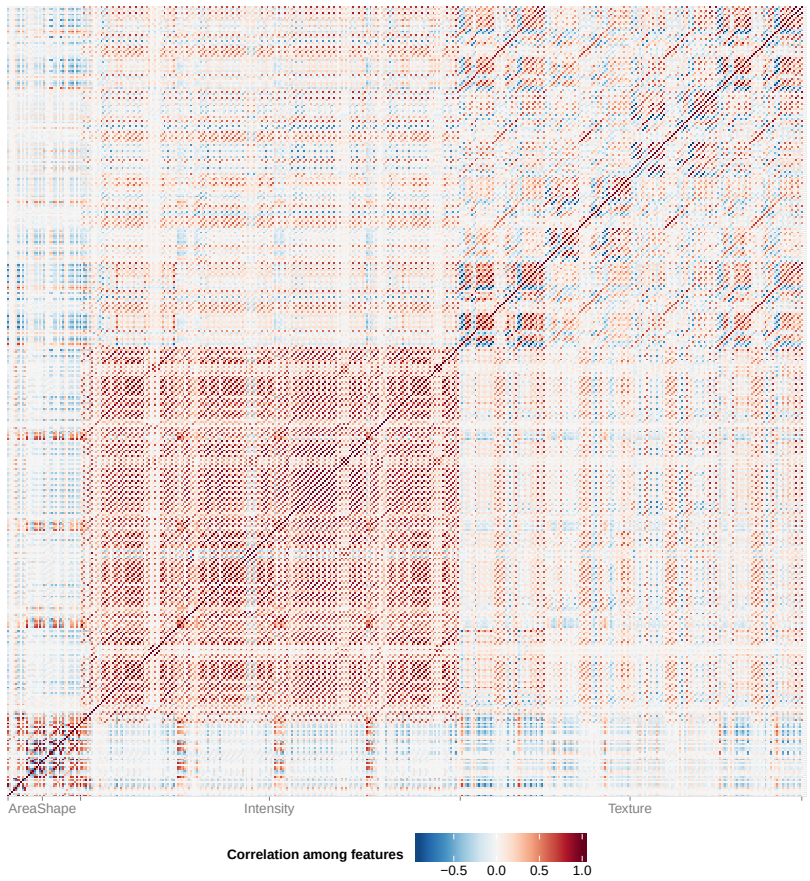


Figure 5.2: A heatmap representation of the correlation matrix obtained by sampling 10% of single cell feature data available for plate J110-2D illustrates severe correlation among many features that is typical for all datasets. This comes as no surprise due to the redundancy in measured features. The three diagonal blocks correspond to three groups of features, *AreaShape*, *Intensity* and *Texture*.

log-likelihood l, as well as a penalty term for model size k, and is defined as

$$\mathrm{AIC} = 2k - 2l(\widehat{\pi}; y). \tag{5.28}$$

Deviance values require caution when interpreted as a goodness-of-fit criterion, especially when used as an absolute measure, rather than being employed in a comparative capacity for nested models (analysis of deviance). The glm function of the R stats package reports both null deviance, defined as

$$D^{(0)}(y; \pi^{(0)}) = 2l(\widetilde{\pi}; y) - 2l(\pi^{(0)}; y), \tag{5.29}$$

and residual deviance as

$$D(y; \widehat{\pi}) = 2l(\widetilde{\pi}; y) - 2l(\widehat{\pi}; y). \tag{5.30}$$

The saturated model $\widetilde{\pi}$ contains as many parameters as there are data observations (n) and consequently represents the maximally possible likelihood. The null model $\pi^{(0)}$ contains only an intercept term (i.e. $y = \mathit{constant}$), while the proposed model $\widehat{\pi}$ attempts to explain the data using $p + 1$ parameters (one for each covariate and an intercept). Finally, the values reported in table 5.2 as $\Delta_{\mathrm{deviance}}$ are obtained as

$$\Delta_{\mathrm{deviance}} = D^{(0)}(y; \pi^{(0)}) - D(y; \widehat{\pi}), \tag{5.31}$$

therefore describing the difference between the quality of fit of the null model to that of the estimated model. Similarly, for the degrees of freedom,

$$\Delta_{\mathrm{df}} = \mathrm{df}_{\mathrm{null}} - \mathrm{df}_{\mathrm{res}} = n - 1 - (n - (p + 1)) = p. \tag{5.32}$$

Under certain assumptions,[3] $\Delta_{\mathrm{deviance}} \sim \chi^2_p$, and therefore a p-value for the significance of the model fit can be calculated. This is not reproduced, as all fits provide highly significant evidence against the null hypothesis, which assumes the fitted model to be no better than the null model.

[3] Deviance is only distributed as χ^2 in the limit where for each $i \in \{1, 2, \ldots, n\}$, the number of identical covariate rows x_i grows to infinity. For continuous regressors, this is typically not the case, as the number of unique x_i will often be very close to n. In case of binomially distributed response, the possibility of over-dispersion causes additional issues, which are not further described, as this does not apply to the current situation. Nevertheless, Nelder and Wedderburn state that "[t]he χ^2 approximation is usually quite accurate for differences of deviances even though it is inaccurate for the deviances themselves."

Table 5.2: Summaries of several GLM models obtained by pairing wells corresponding to the genes MTOR (H6), PIK3R3 (K8), RIPK4 (G17) and TGFBR1 (M4) with all available scrambled wells on the same plate (J110-2D). Comparisons among scrambled wells serve as baseline (the scrambled row corresponds to well G1). Model fit is summarized by AIC, the difference in deviance and degrees of freedom (both between null and fitted models), as well as prediction scores. The following models suffer from separated data: MTOR (A24, G1, J2), PIK3R3 (E24, G1, G23, H2, L23), RIPK4 (E2, G23, H2, J2, L1) and Scrambled (A24, G23, H24, J24, L23).

		A2	A24	E2	E24	G1	G23	H2	H24	J2	J24	L1	L23
MTOR	AIC	1482	3258	1639	3210	1386	3546	1590	3120	1480	3243	1922	3762
	Δ_{deviance}	3353	1726	3167	2178	3572	1427	2773	1776	3439	1531	2927	1396
	Δ_{df}	46	48	48	47	46	48	48	48	48	48	47	48
	Acc	0.92	0.77	0.9	0.81	0.91	0.75	0.9	0.79	0.92	0.79	0.89	0.76
	Mcc	0.84	0.54	0.79	0.61	0.82	0.49	0.79	0.58	0.84	0.57	0.78	0.51
PIK3R3	AIC	4513	4425	4027	4961	3573	3646	3452	4377	3734	3201	4198	3258
	Δ_{deviance}	840	1100	1292	1037	1926	1865	1353	1044	1717	2080	1171	2468
	Δ_{df}	48	49	50	49	49	49	49	49	50	49	49	49
	Acc	0.71	0.74	0.72	0.71	0.79	0.77	0.75	0.72	0.76	0.81	0.73	0.84
	Mcc	0.43	0.48	0.44	0.4	0.57	0.54	0.49	0.44	0.52	0.63	0.46	0.68
RIPK4	AIC	4012	5498	3548	6013	3117	4830	3134	5045	3321	4503	4028	4728
	Δ_{deviance}	1803	510	2227	532	2860	1163	2064	846	2604	1230	1806	1508
	Δ_{df}	48	49	49	49	48	49	49	49	50	49	49	49
	Acc	0.77	0.63	0.81	0.59	0.85	0.7	0.82	0.65	0.84	0.72	0.77	0.74
	Mcc	0.54	0.25	0.61	0.18	0.7	0.41	0.63	0.28	0.68	0.42	0.52	0.48
Scrambled	AIC	4617	2405	4396	2753	–	2186	3707	2866	4559	1725	4702	1938
	Δ_{deviance}	726	3106	911	3230	–	3312	1089	2544	880	3541	655	3775
	Δ_{df}	48	48	49	48	–	48	49	49	49	47	48	48
	Acc	0.66	0.88	0.7	0.84	–	0.9	0.73	0.87	0.66	0.91	0.67	0.92
	Mcc	0.32	0.77	0.41	0.68	–	0.8	0.46	0.73	0.32	0.83	0.33	0.84
TGFBR1	AIC	3842	4935	3796	5522	3745	4850	3261	4998	3542	3838	4282	4676
	Δ_{deviance}	2073	1179	2080	1142	2339	1248	2024	996	2487	1993	1652	1672
	Δ_{df}	47	48	49	48	48	48	49	48	49	48	48	48
	Acc	0.79	0.71	0.79	0.69	0.81	0.73	0.83	0.71	0.83	0.79	0.76	0.73
	Mcc	0.58	0.4	0.58	0.37	0.62	0.45	0.64	0.42	0.66	0.58	0.51	0.46

Moving along to prediction scores, table 5.2 shows both accuracy and Matthews correlation coefficient (MCC) values obtained by separating 20% of data for each group from training data and evaluating predictions using test data. Accuracy is defined as

$$\mathrm{Acc} = \frac{n_{tp} + n_{tn}}{n_p + n_n} \tag{5.33}$$

where n_{tp} represents the number of true positives, n_{tn} the count of true negatives and p, n the number of positive and negative instances, respectively. The MCC (Matthews 1975) can be evaluated as

$$\mathrm{Mcc} = \frac{n_{tp}n_{tn} - n_{fp}n_{fn}}{\sqrt{(n_{tp} + n_{fp})(n_{tp} + n_{fn})(n_{tn} + n_{fp})(n_{tn} + n_{fn})}} \tag{5.34}$$

and n_{fp}, n_{fn} correspond to false positives and false negatives. Values for MCC range from -1 (total disagreement) to 1 (perfect prediction), and the midpoint 0 indicates random prediction.

Given these model characteristics, two patterns emerge: (1) the ability to distinguish two wells is dependent on well distance within the plate and (2) the differences among scrambled wells, in terms of computed quality of fit estimates, are comparable to those that are observed when modeling the discrepancy between hit genes and control wells. To make the first claim, well locations are required: MTOR (H6), PIK3R3 (K8), RIPK4 (G17), TGFBR1 (M4) and Scrambled (A2, A24, E2, E24, G1, G23, H2, H24, J2, J24, L1 and L23). For both MTOR and TGFBR1, which represent early-row, down-hit genes, an alternating sequence is clearly discernible and is characterized by lower AIC, larger Δ_{deviance} and better predictive power for wells that are closer together, while the opposite holds for comparisons among scrambled wells. The effect is less distinct for PIK3R3 and RIPK4 which both are located more towards the plate center and are identified as up-hits. For RIPK4 the alternations are still noticeable albeit, as in scrambled, polarity is reversed. This is indicative of some technical artifacts contained in the data, that dominate biological features of interest. Furthermore, the excellent predictions that can be made based on membership to either one of a scrambled well pair is disconcerting, as biologically they should be equivalent. Again, technical effects seemingly dominate.

In 18 of the 59 models displayed in table 5.2, a warning regarding perfectly separated data is issued by the glm routine (in the example of MTOR, wells A24, G1 and J2). For this particular case, the matter is not further investigated,

as it does not have obviously relevant consequences.[4] Affected data points appear in line with well pairs where no complete data separation is possible and the above arguments still hold if possibly questionable data is excluded (albeit patterns are less clearly distinguishable).

Apart from the results shown, many similar investigations were performed, using other datasets and/or slightly different methods. Further *Brucella* plates were considered, pooled siRNA experiments, libraries from Ambion and Qiagen, as were several *Salmonella* plates and for some inquiries, cell population was limited to infected only. Method-wise, ridge and elastic net penalized regression (glmnet), other glm implementations, such as glm2 (Marschner 2011), which uses a more robust fitting procedure, brglm (Kosmidis 2007), which deals with data separation by penalized maximum likelihood and bayesglm (Gelman and Hill 2007), capable of regularizing coefficients though a weakly informative prior distribution, as well as step-wise model building (using the step function of the R stats package), was explored.

All analysis performed clearly indicates that for the intended type of modeling, an effective normalization scheme has to be developed that is able to capture technical effects (and perhaps even spurious biological artifacts), without destroying phenotypic information coming from gene knockdown and pathogen infection. Attempts of achieving this are outlined in the following section.

5.3 Data Normalization

The large amount of technical variation, coupled with treating biological systems, which in turn are associated with their own inherent noise, make the analysis of HTS data a challenging endeavor, requiring thorough normalization methods. The poor reproducibility that often afflicts siRNA experiments may in part be addressed and even resolved with the development of effective corrections that do not negatively affect phenotypes of interest. The following sections outline two types of normalization approaches, plate and well level corrections via Z-scoring/B-scoring and using residuals of a MARS model

[4] For other inquiries, not reported here, perfect separation was addressed by reducing the number of PCs or using regularized procedures. Furthermore the issue was studied by solving associated linear programming problems in order to explicitly find the separating hyperplane and thus determine whether this is actually part of the data or caused by numerical shortcomings of the glm implementation. In an example by Gelman and Hill it is shown that a response such as `y <- rep (c(1,0),c(10,5))` in a model containing only an intercept, will trigger the separation warning dependent on starting values (i.e. using `start=2.6`, glm runs fine, whereas `start=2.7` issues the warning). A further effect known to cause problems under certain circumstances, especially for convergence, is the Hauck-Donner phenomenon. As it turns out, separating hyperplanes can reliably be determined, most probably owing in part to the high dimensional setting (with respect to predictor variables).

at single cell level, as well as the application of such procedures to InfectX datasets.

5.3.1 Plate and Well Level Normalization

In order to correct siRNA data for experimental artifacts at plate and well levels, two schemes have become standard practice, Z-scoring and B-scoring (Malo et al. 2006). The former is widely known (outside the field of siRNA analysis) and is defined as

$$z_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j}, \quad i = 1, \ldots, n \text{ and } j = 1, \ldots, m \tag{5.35}$$

where n is the number of cells, m the number of features and $x_{.j}$ the data vector of feature j to be normalized, while μ_j represents the sample mean and σ_j the sample standard deviation of feature j, as

$$\mu_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij} \quad \text{and} \quad \sigma_j = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_{ij} - \mu_j)^2}. \tag{5.36}$$

Applying Z-scoring therefore both centers data around zero and scales dispersion to unit variance. B-scoring is more domain specific and deals with row and column effects that have been discussed previously (e.g. pipetting issues, leading to horizontal patterns or temporal effects such as decay of actin stain intensity, resulting in a vertically oriented gradient, when imaging is performed column-wise). B-scoring can be expressed as

$$b_{rcp} = \frac{\varepsilon_{rcp}}{\text{mad}(r)} = \frac{x_{rcp} - (\widehat{\mu}_p + \widehat{\alpha}_{rp} + \widehat{\beta}_{cp})}{\text{mad}(r)}, \quad r = 1, \ldots, N \text{ and } c = 1, \ldots, M, \tag{5.37}$$

where N is the number of plate rows and M the number of plate columns (in the present setup 16 and 24 respectively). Estimates for row and column effects, $\widehat{\alpha}_{rp}$ and $\widehat{\beta}_{cp}$, are obtained by fitting a two-way median polish algorithm. Together with an estimate for plate average $\widehat{\mu}_p$, these three parameters are used to determine a residual ε_{rcp}, which divided by the median absolute deviation (MAD) over the whole plate, yields the B-scored value b_{rcp}. Median polishing proceeds by augmenting the plate layout with an additional column and row as

$$\begin{array}{cccc|c} \varepsilon_{1,1,p} & \varepsilon_{1,2,p} & \cdots & \varepsilon_{1,M,p} & \alpha_{1,p} \\ \varepsilon_{2,1,p} & \varepsilon_{2,2,p} & \cdots & \varepsilon_{2,M,p} & \alpha_{2,p} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \varepsilon_{N,1,p} & \varepsilon_{N,2,p} & \cdots & \varepsilon_{N,M,p} & \alpha_{N,p} \\ \hline \beta_{1,p} & \beta_{2,p} & \cdots & \beta_{M,p} & \mu_p \end{array}$$

and initializing the values as $\varepsilon_{rcp} = x_{rcp}$ and $\alpha_{rp} = \beta_{cp} = \mu_p = 0$. A row sweep consists of iterating all rows, calculating the median of $(\varepsilon_{i,1,p}, \varepsilon_{i,2,p}, \ldots, \varepsilon_{i,M,p})$ for each row i, subtracting the resulting value from $(\varepsilon_{i,1,p}, \varepsilon_{i,2,p}, \ldots, \varepsilon_{i,M,p})$ and adding it to $\alpha_{i,p}$. The same procedure is also applied to the column effect row where the median of $(\beta_{1,p}, \beta_{2,p}, \ldots, \beta_{M,p}$ is subtracted from $(\beta_{1,p}, \beta_{2,p}, \ldots, \beta_{M,p}$ and added to $\mu_p)$. A column sweep is carried out analogously and the two procedures are alternated until all rows and columns of residuals have median zero, as do the vectors of row and column effects (or fall below a threshold close to zero). Usually only a couple of sweeps in each direction (~2) are needed. The results are non-unique as they depend on whether row or column sweeps are put first. Furthermore, using means instead of medians yields a least squares decomposition as in two-way ANOVA without iteration, which is less robust towards outliers (Brown 2006; Venables and Ripley 2002). The MAD is defined as

$$\text{mad}(x) = \text{median}\left(|x_k - \text{median}(x)|\right). \tag{5.38}$$

and therefore provides an estimation of data spread which is more robust towards outliers as other measures of dispersion, such as standard deviation.

While B-scoring has proven to be a capable normalization tool for data coming from a plate reader or phenotypic data like infection scores as generated from InfectX screens, where there is a single value per well to be adjusted for experimental artifacts, the situation for single cell feature data is much more complex. Ideally, Z-scoring and B-scoring are able to correct for issues at plate and well level, but data hierarchy goes beyond that for InfectX datasets. For example, data can be split into images which are captured individually, possibly with different imaging parameters and therefore may require differing treatment.

5.3.2 Multivariate Adaptive Regression Splines

At cellular level of granularity, an abundance of additional sources of noise may directly be addressed. This includes technical issues, as well as biological

variability. Examples for the former are location of cell within the well which might be relevant due some degree of curvature of the well bottom, inducing focusing problems towards well borders, or location of cell within image, possibly affecting cellular features due to varying optical properties moving away from the image center (e.g. vignetting, decrease in sharpness, chromatic aberration, etc.). Biological sources of noise are even more plentiful and therefore increasingly harder to address. Obvious targets include general cell state parameters, such as cell cycle stage or whether the cell is apoptotic with the difficulty here being reliably determining these factor variables.

Work by Snijder et al. elucidates the importance of cellular population context in cell-to-cell variability. In a comprehensive analysis of virally perturbed siRNA screens they demonstrate that parameters such as local cell density, population size and cell location within cellular aggregates significantly alter measured phenotypes. Building on these results, Knapp et al. propose a normalization scheme by fitting a MARS model to a selection of features that represent the cellular population context (among other technical parameters) and using only the residuals for further analysis.

MARS is an nonparametric regression procedure for finding a piecewise linear solution. No assumptions on data distributions are made and MARS represents a capable method for high-dimensional settings with respect to predictor variables. The following short introduction into MARS modeling is largely based on Hastie, Tibshirani, and Friedman (2009). Basis functions of the form

$$(x_j - t)_+ = \max(0, x_j - t) = \begin{cases} x_j - t, & \text{if } x_j > t \\ 0, & \text{otherwise,} \end{cases} \tag{5.39}$$

and $(t - x_j)_+$, which are combined as reflected pairs, are used for describing the model surface. Such linear splines contain a knot at value t, separating the function support into a zero part and a nonzero domain, which for the reflected version are swapped with opposite slope. Despite each basis function only depending on a single covariate ($j \in \{1, 2, \ldots, p\}$), they are considered as functions over the complete predictor space $\mathbb{R}^p$. Model building proceeds by maintaining two sets of reflected pairs, candidates and active pairs, where the active set initially contains only a constant term and candidates include all $2np$ possible functions with knots at each observed value x_{ij}

$$C = \left\{ (x_j - t)_+, (t - x_j)_+ \mid t \in \{x_{1j}, x_{2j}, \ldots, x_{nj},\} \wedge j \in \{1, 2, \ldots, p\} \right\}. \tag{5.40}$$

A MARS model has the form

$$g(x) = \mu + \sum_{m=1}^{M} \beta_m h_m(x), \tag{5.41}$$

where the coefficients β_m represent slopes of basis functions $h_m(\cdot)$ which can either be chosen from individual functions in C or by forming products of functions from the set of candidates C and thereby directly model interactions between variables. The active set is initialized with $A = \{h_0(x) = 1\}$ and in an iterative procedure, for $k = 1, 2, \ldots, M$, the best pair of functions $\{h_{2k-1}(x), h_{2k}(x)\}$ with respect to the largest reduction in residual sum of squares is chosen and added to the active set, whereby the new additions are products of a reflected pair from the candidate set with a function $h_l(x)$ of the active set

$$h_{2k-1}(x) = h_l(x) \cdot (x_j - t)_+ \tag{5.42a}$$
$$h_{2k}(x) = h_l(x) \cdot (t - x_j)_+. \tag{5.42b}$$

The $2k$ coefficients are estimated by least squares and in each step the model grows by two basis functions. Termination of the iteration process occurs when a preset number of basis functions have been added to the active set, typically leading to a model that over-fits the data. A pruning scheme follows that from each pair of functions $\{h_{2k-1}(x), h_{2k}(x)\}$, removes the one that yields the smaller increase in residual sum of squares. In order to determine the right amount of backwards elimination and find the best model $\widehat{g}_\lambda^*(x)$, for each stage, a generalized cross validation (GCV) score is computed for the current model $\widehat{g}_\lambda(x)$. Another possibility is performing cross validation but this is often foregone due to computational expense. The GCV criterion is defined as

$$\mathrm{gcv}(\lambda) = \frac{1}{n} \frac{\sum_{i=1}^{n} \left(y_i - \widehat{g}_\lambda(x_i)\right)^2}{\left(1 - \frac{C(\lambda)}{n}\right)^2}, \tag{5.43}$$

where cost complexity term $C(\lambda)$ represents the effective number of parameters, computed as a sum of the number of linearly independent basis functions with the product of a smoothing parameter d and the total number of terms. The value of d is typically 2 (additive model) or 3 (higher orders allowed) and controls the amount of kinks introduced (Friedman 1991).

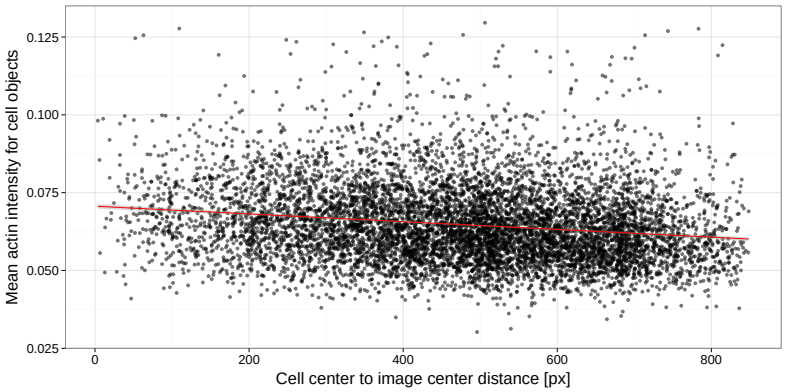


Figure 5.3: In order to illustrate the relationship between object location and feature values, a thinned out scatterplot (only a randomly sampled 1% of datapoints are shown) of mean actin intensity against distance from image center is reproduced alongside a trend-line as calculated by gam of the CRAN mgcv package. An approximately linear trend is discernible which can be found in many feature types.

By having the option of combining functions in the active set with newly entering terms, the algorithm builds a hierarchical model in the sense that interactions of added variables are only possible if all interacting partners are already present, forming an interaction of one order less. The reasoning behind this is that otherwise the search space would grow exponentially, causing computational issues for higher order interactions and in many cases it seems justifiable to require main effects as basis for interactions. Furthermore, the formation of powers is not allowed as a term may only enter a single time in an iteration, again limiting the search space in favor of computational efficiency. For interpretability and in larger problems for performance reasons, it is often advisable to restrict the number of interactions to degree 2 or 3.

5.3.3 Normalization of Single Cell Data

Implemented as part of singleCellFeatures, are both procedures for applying Z-scoring and B-scoring, as well as fitting a user-defined MARS model to each feature selected to be normalized (see section 4.3.1). Two sets of features so far have been used as predictors in MARS, a simpler, more conservative selection intended for targeting technical issues and a larger set based on the work of Knapp et al. (2011), which includes population context.

The smaller of the two contains predictors for object location within image and well, in addition to feature specific terms obtained through B-scoring. Moti-

vated by findings as displayed in figure 5.3, cellular features are normalized using the locations of their respective nuclei. Figure 5.3 shows a scatter-plot of mean actin intensity in cell objects versus their locations measured as Euclidean distance from the image center. The trend line (calculated by the function `gam` of the CRAN package mgcv) indicates an approximately linear relationship with negative slope. Such dependencies can reliably be found for most features, both at well and image level and their significance can be established by fitting multiple linear regression models. The resulting p-values for an overwhelming majority of features are highly significant, below machine accuracy ($< 2 \cdot 10^{-16}$). As this effect constitutes an entirely experimental artifact, it seems reasonable to correct for location with respect to well and image centers.

Population context normalization additionally includes features for nucleus area, cell count per image, nuclear form factor, cell area, cell density, whether the cell is close to an image border and whether the cell is located at the edge of a colony. Knapp et al. suggest that the procedure be carried out for a complete screen and while that is not a difficult task for the data they processed (only a single feature, 10-fold fewer cells), it is far more demanding for IndectX data. For an investigation considering only infected cells of a *Brucella*, Dharmacon unpooled screen, this was necessary as only few cells per MTOR well are available which consequently have to be aggregated.

In order for the MARS procedure to even out technical effects, spanning multiple plates, it is, at least for the dataset mentioned above, important to center data with respect to plate means prior to fitting the MARS model, as otherwise results are unsatisfactory (neither row, nor column effects can be completely removed, most probably due to underestimated plate effects). A further issue is that of memory. The entire screen consists of 2×12 plates which would require on the order of 200 GB for storing the data alone, not taking operational overhead into account. This is infeasible, even for large memory machines. As MTOR wells are located on 8 of the 24 plates, only these were handled jointly.

Computationally, two avenues were explored, keeping all data in memory and utilizing disk scratch to lower the astronomical memory requirements of the former approach. Using a reasonably fast permanent storage device such as a solid state drive, screen wide normalization is readily possible on regular desktop machines but frequent data fetching from storage incurs a significant time penalty. Processing 8 plates and ~400 features takes on the order of 36 h. Handling all data in memory speeds up the process to 6–8 h, but handling 8 plates, which alone only require ~60 GB, required a complete 256 GB node of the ETH Euler cluster for processing.

Both the exclusively technical normalization procedure and the one including biological features additionally incorporate predictors obtained through

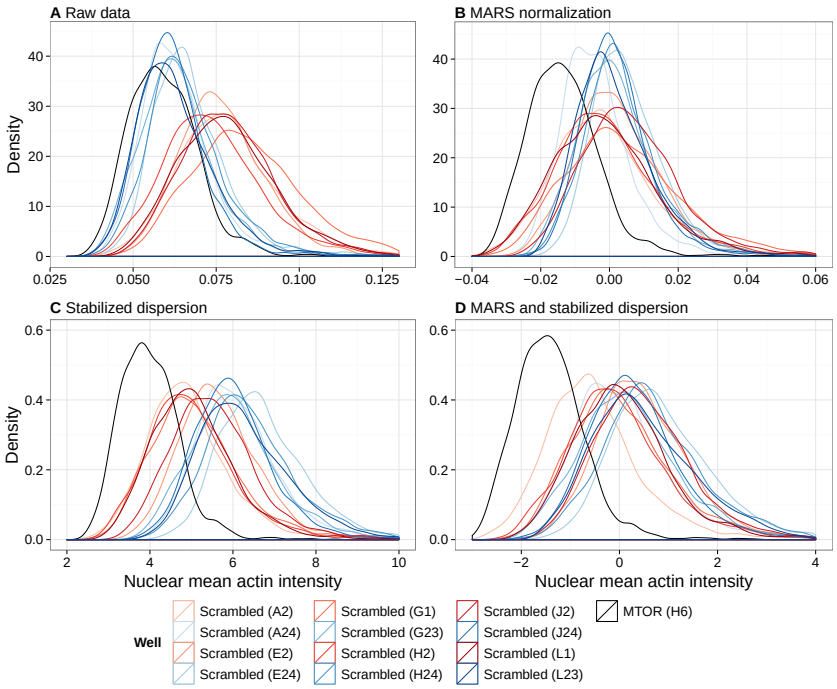


Figure 5.4: In order to provide intuitive access to how normalization affects the data, four density plots are reproduced, using mean nuclear actin intensity for MTOR and scrambled wells of plate J110-2D. The top left plot represents the raw data and both a clear separation of two groups of scrambled wells (corresponding to early and late column wells) can be recognized, as well as the issue that differences between MTOR wells and scrambled wells appear no larger than differences among scrambled wells (A). Moving to the right partially recovers some issues, as scrambled distributions now all roughly share the same center (B), while moving downwards improves the problem of varying dispersion with respect to scrambled grouping (C). Finally, bottom right represents a combination of both schemes, yielding the best result in that scrambled wells become more similar while differences to MTOR are retained (D).

B-scoring. For each of the features that are selected, the corresponding row, column and plate effects are calculated and are included with the mars model which consequently contains 5 predictors for the simple and 9 predictors for the more complex variant.

Figure 5.4 is reproduced for providing some intuition on effects that may be addressed by the proposed normalization schemes. Panel (A) represents raw nuclear mean actin intensity densities of all 12 scrambled wells and the single MTOR well on a plate of the *Brucella*, Dharmacon unpooled screen. Color

coding of scrambled wells is based on the two groups that can easily be distinguished, which most probably are responsible for the alternating pattern of table 5.2. Wells that have a low column index are colored in shades of red, while late column wells are blue and colors grow darker with increasing row index. The density corresponding to MTOR is shown in black.

With MARS normalization, mainly due to B-scoring terms, the distance between the group centers are is drastically reduced without loosing information with respect to MTOR (panel B of figure 5.4). Likewise for variance scaling (Z-scoring without centering), the previously distinct amount of dispersion within the two scrambled groups is equalized without affecting MTOR (panel C). In order to make this step more robust, data is not divided by standard deviation, but by MAD and while being carried out on well level, not only the individual wells are used for calculating MAD, but the two vertical and horizontal neighbors are included as well (for plate borders, two neighboring wells on the same side of the target well are selected).

Finally, the two approaches can be coupled by first stabilizing dispersion, followed by MARS normalization (panel D). For this particular feature, the hybrid approach yields the best results in the sense that discrepancies between scrambled wells are reduced without loosing information on MTOR. Building on these results, one could assume that the combined normalization strategy should considerably improve GLM modeling. However, while MARS succeeds at recovering the data from column dependence, variance scaling does not help. Similarly for the two predictor sets in MARS, the more complicated one does not yield better data quality. Therefore, the more parsimonious normalization procedure constituting only of MARS targeted at technical issues was used to generate table 5.3.

When comparing table 5.3 to 5.2, the most striking difference is the disappearance of the striped pattern due to row dependence. This causes the previously very high prediction accuracies in scrambled wells to drop from ~90% to slightly more reasonable but still high ~70%. Furthermore, a clear difference between MTOR and scrambled rows is discernible. MTOR is consistently characterized by lower AIC (about half), much larger Δ_{deviance} (2–6 fold difference) and better prediction scores, altogether indicating better modeling of the differences between MTOR and scrambled wells than of diversity within scrambled wells. TGFBR1 results differ from scrambled as well (the same observation holds for table 5.2). Accuracies (and therefore MCCs) are generally lower for the control wells, while Δ_{deviance} is 2–3 fold increased in TGFBR1 wells. These observations are, however, put somewhat into perspective by the other 2 genes that do not exhibit behavior that is easily distinguishable from scrambled, possibly hinting at issues with data quality or analysis routines.

Table 5.3: For illustrating the effect of MARS normalization, using the smaller set of predictors targeted at technical issues only, this table is a reiteration of table 5.2, using normalized data instead. All other parameters (i.e. PCA, 90% of variance, etc.) remain. The following models suffer from separated data: MTOR (A24, E2, E24, G1, G23, H2 and J2), PIK3R3 (H2), RIPK4 (A2, E24, G1, G23, H2, J2 and L1), Scrambled (G23 and H2) and TGFBR1 (E2, E24 and H24).

		A2	A24	E2	E24	G1	G23	H2	H24	J2	J24	L1	L23
MTOR	AIC	2016	2187	1655	1458	1757	1448	1517	1268	1288	1736	2128	1860
	Δ_{deviance}	2821	2798	3151	3931	3203	3522	2846	3625	3632	3037	2723	3298
	Δ_{df}	47	48	48	47	47	47	48	47	48	47	48	48
	Acc	0.87	0.87	0.9	0.92	0.91	0.92	0.91	0.93	0.93	0.91	0.88	0.91
	Mcc	0.75	0.74	0.79	0.83	0.82	0.84	0.81	0.85	0.86	0.82	0.77	0.82
PIK3R3	AIC	4548	4095	4400	5149	4814	4833	3764	4143	4516	4371	4670	4839
	Δ_{deviance}	808	1430	920	849	686	680	1043	1278	937	912	701	889
	Δ_{df}	49	49	50	49	50	50	50	49	51	50	50	50
	Acc	0.72	0.77	0.67	0.67	0.68	0.68	0.76	0.76	0.69	0.72	0.67	0.68
	Mcc	0.45	0.54	0.35	0.33	0.35	0.35	0.5	0.52	0.38	0.44	0.33	0.36
RIPK4	AIC	4929	4862	5025	5995	5217	5303	4050	4677	5037	5221	5156	5574
	Δ_{deviance}	889	1146	751	550	764	692	1149	1214	891	514	679	665
	Δ_{df}	49	49	50	49	50	50	50	49	51	50	50	50
	Acc	0.69	0.72	0.67	0.64	0.69	0.64	0.76	0.75	0.71	0.63	0.68	0.67
	Mcc	0.36	0.44	0.32	0.29	0.36	0.28	0.49	0.5	0.41	0.24	0.34	0.34
Scrambled	AIC	4495	4296	4699	5306	–	4951	4003	4921	5024	4462	4725	4740
	Δ_{deviance}	848	1217	607	679	–	549	795	491	416	809	634	976
	Δ_{df}	48	49	49	49	–	49	50	50	50	49	49	50
	Acc	0.69	0.73	0.65	0.64	–	0.66	0.67	0.66	0.58	0.66	0.62	0.72
	Mcc	0.38	0.45	0.3	0.28	–	0.31	0.33	0.31	0.17	0.32	0.25	0.44
TGFBR1	AIC	4414	4920	4105	4502	4429	4268	3250	3513	3817	4284	4709	4801
	Δ_{deviance}	1504	1194	1770	2162	1655	1832	2035	2481	2212	1549	1227	1549
	Δ_{df}	48	48	49	48	48	49	49	48	49	49	49	49
	Acc	0.76	0.71	0.78	0.77	0.76	0.76	0.82	0.83	0.82	0.76	0.73	0.74
	Mcc	0.52	0.41	0.54	0.53	0.51	0.51	0.62	0.65	0.63	0.51	0.44	0.47

5.4 Outlook and Conclusion

Unfortunately, the goal that was originally set out to achieve, to find a set of influential features that discriminate single cell data of infected cells between an siRNA experiment targeting a hit gene and a scrambled control experiment, could so far not be accomplished. Such results may provide valuable insight into biological mechanisms as to how the down-regulated gene affects infection patterns, in turn, possibly yielding better understanding of pathogen infectivity in human cells. However, issues surrounding robust data normalization remain, despite much effort and prevent sensible inference from fitted models to be drawn. With current conditioning schemes, discrepancies between control wells persist to such an extent that biologically equivalent data can be separated witch 60–70% accuracy (cf. table 5.3), which is a clear indication of technical effects still being present to an unacceptable degree. Furthermore, the extent of inconsistencies among scrambled control wells is on the same order of magnitude as differences between wells containing siRNA sequences against hit genes (with the exception of MTOR), thereby making any possible list of influential features derived from current data highly questionable.

Issues of reproducibility have been plaguing siRNA based HTS ever since its conception and have recently gained attention with researchers calling for standardization of screening practice and more robust hit scoring. An example of the issue is provided by four RNAi screens investigating HIV infection, performed in 2008 and 2009, three of which employ pooled siRNA duplexes while the fourth uses pooled shRNA reagents. On average, each of the screens reported 300 hit genes and interestingly, there is zero overlap among all four and only three genes are shared throughout the three siRNA screens (Bhinder, Shum, and Djaballah 2014).

A large-scale study provided by Bhinder and Djaballah (2013) corroborates concerns over reproducibility and quality of obtained hits. Out of a reviewed list of 300 published screening experiments, 30 lethality-based screens, performed by different research groups, were selected for comparative investigation. Aggregation of hits implicates a third of the overall genome, suggesting that an unlikely 30% of all genes are essential to cellular survival. When investigating commonalities, not a single gene is reported throughout all screens. Constraining analysis to the 16 siRNA screens (the other 14 use shRNA), leaves two genome-wide and 14 focused studies and for both groups, 90% of reported hits are specific to a single screen (orphan hits), while 4 genes are shared among the two genome wide and none are common to all focused experiments.

Such observations are mainly concerned with comparing the results of distinct investigations performed by separate research groups, possibly using individ-

ual procedures and a large source of variation must be attributed to the current lack of standard practice. Nevertheless, for utilizing siRNA based HTS at its full potential, issues of reproducibility must be resolved. To that end, much effort by InfectX has been dedicated to the development of a robust screening platform, providing as many replicates as possible and making data available publicly (a serious reproducibility issue is the nonavailability of raw data as it prevents others from examining and potentially improving data processing). Nevertheless, many technical artifacts, sources of variation intrinsic to siRNA based experimentation, as well as the noisy nature of biological systems have to be handled.

While a fair amount of scrutiny towards data quality at well level, has led to some normalization procedures being developed for corresponding datasets, less work so far has been dedicated to the single cell level. As demonstrated, a combination of Z-scoring, B-scoring and a MARS model as proposed by Knapp et al. (2011) is not sufficient for making single cell data of distinct wells directly comparable. Whether there are further technical effects that are not removed by this set of methods or biological issues that have not been considered are to blame is an open question. Further, it is unclear if correcting for population context is beneficial to the type of investigation at the heart of this project, as it might be overzealous in that cellular feature information that originates from siRNA perturbations and manifests within population context is possibly removed.

Instead of targeting population context related information, sources of variation such as OTEs might provide more attractive objectives. Off-targeting has been established as an important confounding factor in siRNA screening and it is entirely possible that some of the variability among scrambled control experiments is attributable to unintentional gene silencing. While scrambled seed sequences are specifically engineered not to match any known genes, especially three prime untranslated regions, the limited combinatorial space makes some degree of interaction likely. Furthermore, unspecific response may occur and add to the obfuscation of the actual signal. It might be possible to adapt recent developments in deconvolution of off-target confounded RNAi screens, such as provided by Schmich et al. (2015) to the single cell level.

When only focusing on infected cells, the importance of correcting for OTEs increases as wells of different sequences with the same target have to be aggregated in cases where the number of infected cells per single well is low (as is the case in *Brucella* screens). The same holds for pooled siRNA screens, as several nonidentical sequences with a common target are included in the same experiment. Distinguishing the signal of target knockdown from random perturbations constitutes a valuable improvement in such situations.

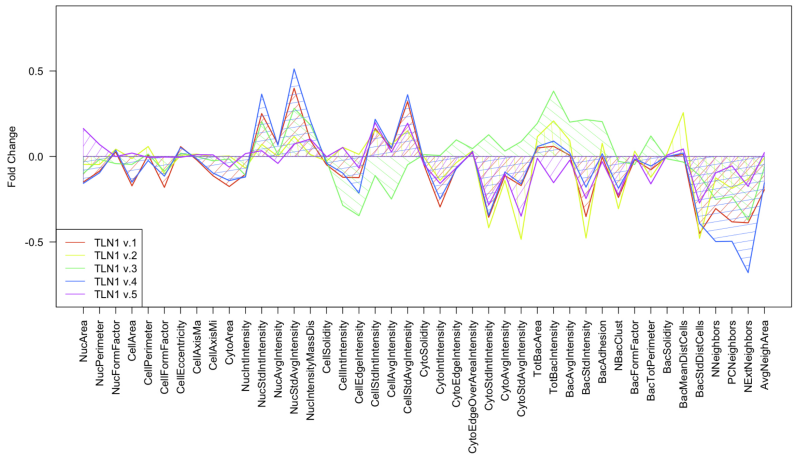


Figure 5.5: A total of 5 distinct siRNA sequences with the shared target talin-1 (TLN1), required for invasome formation in *Bartonella* infection, yield considerable variability in feature space. While there are some commonalities, due to on-target effects, off targeting may cause individual behavior patterns. Figure take from Geier, Truttmann, and Dehio (2010).

In addition to accounting for OTEs, the present normalization scheme could be adapted to treat features individually, instead of repeatedly being applied in the same manner. There is great diversity among the multitude of features that are extracted from imagery which warrants further study. Perhaps individual data transformations need be applied and it may be necessary, albeit labor intensive to customize normalization at feature level. This was conceded but due to limited time availability so far could not be explored further.

If sufficient normalization proves elusive, changing the analysis setup has to be contemplated. Instead of determining influential features by comparing wells, modeling infected versus uninfected cells within a well might yield the desired insights. This alleviates the requirement of directly comparable wells while still determining well specific parameters. The features used for infection scoring are fixed pathogen wide and finding gene specific sets constitutes information that is qualitatively similar to that of the original investigation.

Finally, the number of available features is larger than the number of features considered during analysis and modeling. So far, only cellular features are included that are scalar valued per cell and provide a data point for each cell. Therefore pathogen objects and neighbor features are left out. This additional information could be included by developing appropriate augmentation functions such as location augmentation as implemented by singleCellFeatures. In

that particular case, object coordinates which by themselves do not convey any directly interpretable information are transformed to more useful features, including local object density, distance from center, location within colonies, etc. Similarly, neighbor identity, again by itself not very useful, could be exploited to define object clusters and determine cluster area, cluster size in terms of members, number of infected members, and many other derived features. Adjacency matrices are assembled by singleCellFeatures and await being put to use.

While the desired list of influential features for siRNA perturbed infection so far could not be compiled, a flexible and capable platform for manipulating the large amounts of data associated with single cell level, high throughput screening, was developed. More sophisticated normalization procedures have to be investigated and hopefully, building on lessons learnt from data analysis provided by this project, ways of extracting the initially targeted information from available datasets will be found in the future.

Appendix A

InfectX Protocols

A.1 Materials and Methods for Wet-Lab Procedures

The following sections describe materials and methods employed in pathogen specific protocols. This information has been published in Rämö et al. (2014) and is only reproduced for the reader's convenience.

***B. henselae*-specific protocol.** *Bartonella henselae* ATCC49882^{T} Δ*bep*G containing plasmid pCD353 (M. Dehio et al. 1998) for IPTG-inducible expression of GFP were grown on Columbia base agar (CBA) plates supplemented with 5% defibrinated sheep blood (Oxoid) and 50 µg ml^{-1} kanamycin. Bacteria were incubated at 35 °C in 5% CO_2 for 72 h before re-streaking them on fresh CBA and further growth for 48 h. Cells were washed once after siRNA-transfection with M199 (Invitrogen)/10% FBS using a plate washer (ELx50-16, BioTek). Cells were infected with *B. henselae* at an MOI of 400 in 50 µl of M199/10% FBS and 0.5 mM IPTG (Applichem) and were incubated at 35 °C in 5% CO_2 for 30 h. Fixation at room temperature (rt) was performed using a Multidrop 384 (Thermo Scientific) to wash cells with 50 µl of phosphate-buffered saline (PBS), fixed in 20 µl of 3.7% PFA for 10 min, and washed once more with 50 µl of PBS. Staining was performed on a Biomek liquid handling platform. Fixed cells were washed twice with 25 µl of PBS and blocked in PBS/0.2% bovine serum albumin protein (BSA) for 10 min. Extracellular bacteria were labeled with a rabbit serum 2037 against B. henselae (C. Dehio et al. 1997) and a secondary antibody goat anti rabbit A647 (Jackson Immuno) in PBS/0.2% BSA. Antibodies were incubated for 30 min each and both incubations were followed by two washings with 25 µl of PBS. Cells were then permeabilized with 20 µl of 0.1% Triton X-100 (Sigma) for 10 min and afterwards washed twice with 25 µl of PBS, followed

by the addition of 20 µl of staining solution (PBS containing 1.5 µg ml^{-1} DY-547-Phalloidin, Dyomics and 1 µg ml^{-1} 4′,6-diamidino-2-phenylindole (DAPI), Roche). After 30 min of incubation in the staining solution, cells were washed twice with 25 µl PBS, followed by a final addition of 50 µl of PBS.

***B. abortus*-specific protocol.** *Brucella abortus* 2308 pJC43 (*aphT::GFP*) (Celli, Salcedo, and Gorvel 2005) were grown in tryptic soy broth medium containing 50 µg ml^{-1} kanamycin for 20 h at 37 °C and shaking (100 rpm) to an OD of 0.8–1.1. 50 µl of DMEM/10% containing bacteria was added per well to obtain a final MOI of 10000 using a cell plate washer (ELx50-16, BioTek). Plates were then centrifuged at 400 g for 20 min at 4 °C to synchronize bacterial entry. After 4 h incubation at 37 °C and 5% CO_2, extracellular bacteria were killed by exchanging the infection medium by 50 µl medium supplemented with 10% FBS and 100 µg ml^{-1} gentamicin (Sigma). After a total infection time of 44 h cells were fixed with 3.7% PFA for 20 min at rt with the cell plate washer. Staining was performed using a Biomek liquid handling platform. Cells were washed twice with PBS and permeabilized with 0.1% Triton X (Sigma) for 10 minute. Then, cells were washed twice with PBS, followed by addition of 20 µl of staining solution which includes DAPI (1 µg ml^{-1}, Roche) and DY-547-phalloidin (1.5 µg ml^{-1}, Dyomics) in 0.5% BSA in PBS. Cells were incubated with staining solution for 30 min at rt, washed twice with PBS, followed by final addition of 50 µl PBS.

***L. monocytogenes*-specific protocol.** After washing an overnight culture of *Listeria monocytogenes* EGDe.PrfA*GFP three times with PBS, bacteria were diluted in DMEM supplemented with 1% FBS. Cells were infected at a MOI of 25 in 30 µl infection medium per well. After centrifugation at 1000 rpm for 5 min and incubation for 1 h at 37 °C in 5% CO_2 to allow the bacteria to enter, extracellular bacteria were killed by exchanging the infection medium by 30 µl DMEM supplemented with 10% FBS and 40 µg ml^{-1} gentamicin (Gibco). Both medium exchange steps were carried out with a plate washer (ELx50-16, BioTek). After additional 4 h at 37 °C in a 5% CO_2 atmosphere, cells were fixed for 15 min at rt by adding 30 µl of 8% PFA in PBS to each well using a multidrop 384 device (Thermo Electron Corporation). PFA was removed by four washes with 500 µl PBS per well using the Power Washer 384 (Tecan). Fixed cells were stained for nuclei, actin and bacterially secreted InlC. First, cells were incubated for 30 min with 10 µl per well of primary staining solution (0.2% saponin, PBS) containing rabbit derived anti-InlC serum (1:250). After four washes with 40 µl PBS per well cells were stained with 10 µl per well of the secondary staining solution (0.2% saponin, PBS) containing Alexa Fluor-546 coupled anti-rabbit antibody (1:250, Invitrogen), DAPI (0.7 µg ml^{-1}, Roche), and DY-647-Phalloidin

(2 µg ml^{-1}, Dyomics). After four washes with 40 µl PBS per well, the cells were kept in 40 µl PBS per well. The staining procedure was carried out with a Tecan freedom evo robot.

***S.* typhimurium-specific protocol.** All liquid handing stages of infection, fixation, and immunofluorescence staining were performed on a liquid handling robot (BioTek; EL406). For infection the *S.* typhimurium strain S.Tm^{SopE_pM975} was used. This strain is a single effector strain, only expressing SopE out of the main four SPI-1 encoded effectors (SipA, SopB, SopE2 and SopE). Additionally this strain harbors a plasmid (pM975) that expresses GFP under the control of a SPI-2 (SsaG)-dependent promotor. The bacterial solution was prepared by cultivating a 12 h culture in 0.3 M lysogeny broth (LB) medium containing 50 µg ml^{-1} streptomycin and 50 µg ml^{-1} ampicillin. Afterwards a 4 h subculture (1:20 diluted from the 12 h culture) was cultivated in 0.3 M LB medium containing 50 µg ml^{-1} streptomycin, which reached an $OD_{600\,nm} \approx 1.0$ after the respective 4 h of incubation time. To perform the infection, 16 µl of diluted *S.* typhimurium (MOI of 80) were added to the HeLa cells. After 20 min of incubation at 37 °C and 5% CO_2, the *S.* typhimurium-containing media was replaced by 60 µl DMEM/10% FBS containing 50 µg µl^{-1} streptomycin and 400 µg µl^{-1} gentamicin to kill all remaining extracellular bacteria. After additional 3 h 40 min incubation at 37 °C and 5% CO_2, cells were fixed by adding 35 µl 4% PFA, 4% sucrose in PBS for 20 min at rt. The fixation solution was removed by adding 60 µl PBS containing 400 µg ml^{-1} gentamicin. Cells were permeabilized for 5 min with 40 µl 0.1% Triton X-100 (Sigma-Aldrich). Afterwards 24 µl of staining solution containing DAPI (1:1000, Sigma-Aldrich) and DY-547-phalloidin (1.2 µg ml^{-1}, Dyomics) was added (prepared in blocking buffer consisting of 4% BSA and 4% Sucrose in PBS). After 1 h of incubation at rt, cells were washed three times with PBS followed by the addition of 60 µl PBS containing 400 µg ml^{-1} gentamicin.

***S. flexneri*-specific protocol.** *Shigella flexneri* M90T Δ*vir*G pCK100 (PuhpT::dsRed) were harvested in exponential growth phase and coated with 0.005% poly-L-lysine (Sigma-Aldrich). Afterwards, bacteria were washed with PBS and resuspended in assay medium (DMEM, 2 mM L-Glutamine, 10 mM HEPES). 20 µl of bacterial suspension was added to each well with a final MOI of 15. Plates were then centrifuged for 1 min at 37 °C and incubated at 37 °C and 5% CO_2. After 30 min of infection, 75 µl were aspirated from each well and monensin (Sigma) and gentamicin (Gibco) were added to a final concentration of 66.7 µM and 66.7 µg ml^{-1}, respectively. After a total infection time of 3.5 h, cells were fixed in 4% PFA for 10 min. Liquid handling was performed using the Multidrop 384 (Thermo Scientific) for dispension steps and a plate washer

(ELx50-16, BioTek) for aspiration steps. For immunofluorescent staining, cells were washed with PBS using the Power Washer 384 (Tecan). Subsequently, cells were incubated with a mouse anti-human IL-8 antibody (1:300, BD Biosciences) in staining solution (0.2% saponin in PBS) for 2 h at rt. After washing the cells with PBS, Hoechst (5 µg ml^{-1}, Invitrogen), DY-495-phalloidin (1.2 µg ml^{-1}, Dyomics) and Alexa Fluor 647-coupled goat anti-mouse IgG (1:400, Invitrogen) were added and incubated for 1 h at rt. The staining procedure was performed using the Biomek NXP Laboratory Automation Workstation (Beckman Coulter).

Adenovirus-specific protocol. All liquid handling stages of infection, fixation, and immunofluorescence staining were performed on the automated pipetting system Well Mate (Thermo Scientific Matrix) and washer Hydrospeed (Tecan). For infection screens recombinant Ad2_ΔE3B-eGFP (short Adenovirus) was utilized as described before (Suomalainen et al. 2013; Yakimovich et al. 2012). Adenovirus was added to cells at an MOI of 0.1 in 10 µl of an infection media/FBS (DMEM supplemented with L-glutamine, 10% FBS, 1% Pen/Strep, Invitrogen). Screening plates were incubated at 37 °C for 16 h, and cells were fixed by adding 21 µl of 16% PFA directly to the cells in culture media for 45 min at rt or long-term storage at 4 °C. Cells were washed 2 times with PBS/25 mM NH_4Cl, permeabilized with 25 µl 0.1% Triton X-100 (Pharmaciebiothek). After 2 washes with PBS the samples were incubated at rt for 1 h with 25 µl staining solution (PBS) containing DAPI (1 µg ml^{-1}, Sigma-Aldrich) and DY-647-phalloidin (1 µg ml^{-1}, Dyomics), washed 2 times with PBS and stored until imaging in 50 µl PBS/NaN_3.

Rhinovirus-specific protocol. All liquid handling stages of infection, fixation, and immunofluorescence staining were performed on the automated pipetting system Well Mate (Thermo Scientific Matrix) and washer Hydrospeed (Tecan). For infection assays with human Rhinovirus serotype 1a (HRV1a) were carried out as described, except that the anti-VP2 antibody Mab 16/7 was used for staining of the infected cells as described earlier (Jurgeit et al. 2012, 2010; Mosser et al. 2002). Rhinovirus at an MOI of 8 was added to cells in 20 µl of an infection media/BSA (DMEM supplemented with GlutaMAX, 30 mM $MgCl_2$ and 0.2% BSA, Invitrogen). Screening plates were incubated for 7 h at 37 °C, and cells were fixed by adding 33 µl of 16% PFA directly to the culture medium. Fixation was either for 30 min at rt or long term storage at 4 °C. Cells were washed twice with PBS/25 mM H_2O, permeabilized with 50 µl 0.2% Triton X-100 (Sigma- Aldrich) followed by 3 PBS washes and blocking with PBS containing 1% BSA (Fraction V, Sigma-Aldrich). Fixed and permeabilized cells were incubated at rt for 1 h with diluted mabR16-7 antibody (0.45 µg ml^{-1}) in

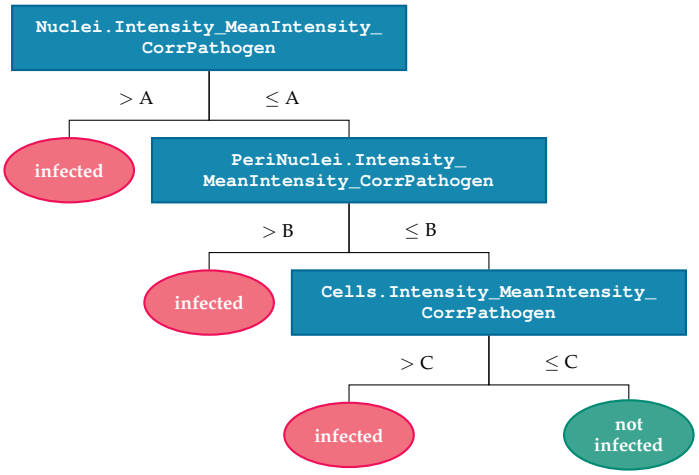
PBS/1% BSA. Cells were washed 3 times with PBS and incubated with 25 µl secondary staining solution (PBS/1% BSA) containing Alexa Fluor 488 secondary antibody (1 µg ml^{-1}, Invitrogen), DAPI (1 µg ml^{-1}, Sigma-Aldrich), and DY-647-phalloidin (0.2 µg ml^{-1}, Dyomics). Cells were washed twice with PBS after 2 h of incubation in secondary staining solution and stored in 50 µl PBS/NaN_3.

Vacciniavirus-specific protocol. All liquid handing stages of infection, fixation, and immunofluorescence staining were performed on a liquid handling robot (BioTek, EL406). For infection assays a recombinant WR VACV, WR E EGFP/ L mCherry, was utilized. For infection, media was aspirated from the RNAi-transfected cell plates and replaced with 40 µl of virus solution per well (MOI of 0.125). Screening plates were incubated for 1 h at 37 °C to allow for infection, after which virus-containing media was removed and replaced with 40 µl DMEM/10% FBS. 8 h after infection 40 µl of DMEM/10% FBS containing 20 µM cytosine arabinoside (AraC) was added to all wells to prevent virus DNA replication in secondary infected cells. 24 h after infection cells were fixed by the addition of 20 µl 18% PFA for 30 min followed by two PBS washes of 80 µl. For immunofluorescence staining of EGFP, cells were incubated for 2 h in 30 µl primary staining solution (0.5% Triton X-100, 0.5% BSA, PBS) per well, containing anti-GFP antibody (1:1000). Cells were washed twice in 80 µl PBS, followed by the addition of 30 µl secondary staining solution (0.5% BSA, PBS) containing Alexa Fluor 488 secondary antibody (1:1000), Hoechst (1:10000), and DY-647-phalloidin (1:1200, Dyomics). Cells were washed twice with 80 µl PBS after 1 h incubation in secondary staining solution followed by the addition of 80 µl H_2O.

A.2 Decision Trees for Infection Scoring

The decision trees for adenovirus and *Bartonella* are shown in section 3.5 while the ones corresponding to the remaining pathogens (*Brucella*, *Listeria*, rhinovirus, *Salmonella* and vaccinia virus) follow[1]. Please refer to section 3.5 for more information of infection scoring, including descriptions of infection patterns upon which these decision trees are based. The corresponding tables show the different thresholds that currently are in use. Due to several experimental parameters affecting intensity measurements, such as age of microscope lamp, plate position in imaging queue or quality of staining, plate-wise adjustments are necessary in some instances. Several plates can be subjected to the same values, but over all datasets, some variation exists. Accompanying each of the following decision trees is a table holding the currently used sets of decision

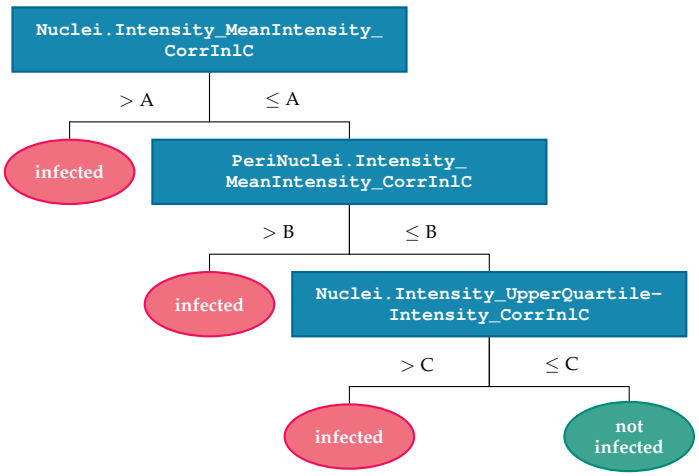
[1] *Shigella* infection scoring currently does not rely on DTIS, therefore no decision tree is shown. Please refer to section 3.5 for details on detection of infected cells in *Shigella* screens.



A	0.050	0.050	0.060	0.075	0.045	0.050	0.040	0.055	0.050	0.040	0.050	0.040
B	0.055	0.055	0.065	0.090	0.050	0.055	0.045	0.060	0.055	0.045	0.055	0.045
C	0.075	0.070	0.085	0.105	0.070	0.070	0.075	0.080	0.075	0.065	0.070	0.065
Abundance	31%	9%	9%	8%	7%	4%	3%	3%	3%	3%	3%	3%

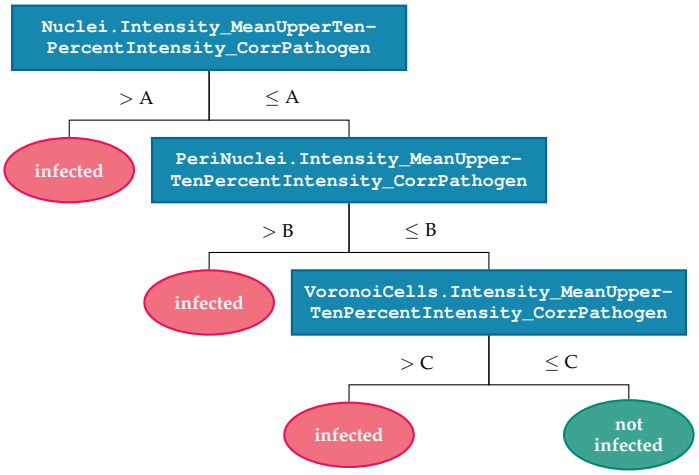
Figure A.1: Decision tree for *Brucella* infection scoring. While the first two decisions are modeled to capture what is considered a normal infection pattern, the last split imposes a high threshold for cells that have failed the first two steps to still be considered infected. The list of decision boundaries values is not exhaustive and the remaining 14% of plates is handled by an additional 29 parameter sets.

boundaries alongside a percentage value indicating the coverage of a given parameter set. For *Brucella* and vaccinia, only the 12 most frequent sets are printed, while the lists for the remaining pathogens are exhaustive.



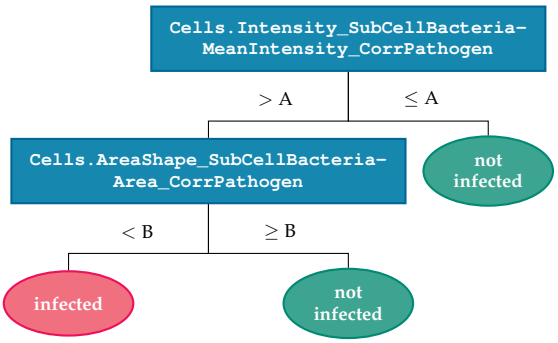
A	0.11	0.29	0.28	0.21	0.20	0.32	0.30	0.22	0.19
B	0.15	0.36	0.35	0.28	0.27	0.39	0.37	0.29	0.26
C	0.12	0.31	0.30	0.23	0.22	0.34	0.32	0.24	0.21
Abundance	25%	12%	12%	12%	12%	6%	6%	6%	6%

Figure A.2: The decision tree for *Listeria* infection scoring is based on a channel recording InlC localization and intensity instead of targeting the bacteria themselves. Despite the values indicating coverage of each of the parameter sets not summing to unity, the list is exhaustive. The discrepancy is caused by rounding.



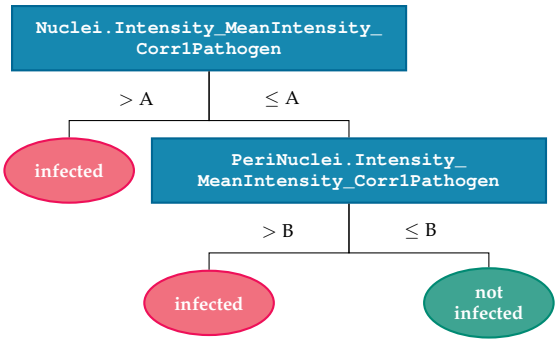
A	0.095	0.090	0.085	0.115	0.080
B	0.100	0.095	0.090	0.125	0.085
C	0.140	0.130	0.125	0.165	0.120
Abundance	45%	35%	14%	5%	1%

Figure A.3: Decision tree for rhinovirus infection scoring. Using the mean of the uppermost decile of pathogen channel intensity data yields the most stable results.



A	0.095	0.100	0.068	0.070	0.082	0.085	0.090	0.067	0.105	0.080	0.065	0.069
B	2	2	2	2	2	2	2	2	2	2	2	2
Abundance	41%	11%	9%	7%	7%	6%	6%	6%	2%	2%	2%	2%

Figure A.4: Decision tree for *Salmonella* infection scoring. For a cell being considered infected, not only does the threshold for pathogen intensity throughout the cell need be exceeded but the bacteria also have to be sufficiently aggregated. The list of decision boundaries is exhaustive and the sum of coverage percentages overshooting unity is caused by rounding.



A	0.030	0.040	0.038	0.035	0.045	0.030	0.030	0.040	0.030	0.055	0.030	0.035
B	0.030	0.045	0.042	0.040	0.050	0.032	0.032	0.045	0.030	0.060	0.035	0.040
Abundance	12%	12%	12%	12%	5%	5%	4%	4%	4%	3%	3%	3%

Figure A.5: Decision tree for vaccinia virus infection scoring. A separate decision tree for distinguishing primary from secondary infections has been developed but is not shown. Not all decision boundary values are shown (the remaining 21% of plates is handled by an additional 21 parameter sets).

Appendix B

SingleCellFeatures Manual

This appendix complements chapter 4, which focuses of architectural and implementation aspects of the presented R package, with some practical guidance on how to install and maintain singleCellFeatures. Furthermore, some examples of how various functionality provided by singleCellFeatures can be used in practice will be given.

B.1 Package Installation

All R code is available on github and can be directly installed from an R session through the devtools package. External requirements are pigz, a parallel implementation of gzip and access to an openBIS (Bauch et al. 2011) instance via the corresponding Java command line tool, which has to be compiled and installed locally.

```
install.packages("devtools")
library(devtools)

install_github("nbenn/singleCellFeatures")
```

Alternatively the package can be downloaded and installed manually by running the following commands in a shell (dependent on where the zip file was downloaded to).

Listing B.1: In order for singleCellData to be correctly configured for a given system, several settings can be adjusted through a yaml-based configuration file.

```
dataStorage:
  dataDir: "path/to/data/dir"
  metaDir: "path/to/metadata/dir"
beeDownloader:
  executable: "path/to/trunk/openBIS/Tools/BeeDataSetDownloader"
  beeSoftsrc: "path/to/trunk"
openBIS:
  username: "user"
  password: "password"
singleCellFeatures:
  sourceDir: "path/to/source"
```

```
unzip ~/Downloads/singleCellFeatures-master.zip
R CMD INSTALL --no-multiarch --with-keep.source \
  ~/Downloads/singleCellFeatures-master
```

Some setup dependent information has to be provided, all of which is stored in a yaml formatted configuration file. The default location of this config file is `~/.singleCellFeaturesConfig`. This can be changed on a per-session basis using the function `configPathSet()` of more permanently, using an `.Rprofile` file. In addition, singleCellFeatures provides a function to generate a template file that can be edited to suit the current setup.

```
## if no config file is present, set one up
# set the config file location
configPathSet("path/to/where/you/want/your/config.yaml")
# create a template file
configInit()
# using a text editor, modify this file for your system

## for inter-session persistence, add the following to your .Rprofile
options(singleCellFeatures.configPath = "path/to/your/config.yaml")
```

The configuration file structure is shown in listing B.1. The two entries under `dataStorage` specify local file-system paths to be used for storing downloaded data and metadata. The first location (`dataDir`) should be chosen such that it points to a location on a volume with several GBs of free storage, in order to be able to hold a couple of plates. A complete plate requires 1–2 GB of storage and having upwards of 50 GB available is recommended. The second path

(`metaDir`) specifies the location of metadata files, the generation of which is described in section B.3. The section `beeDownloader` is concerned with the location of the Java command line tool for accessing openBIS data, which in case of InfectX is called BeeDataSetDownloader. Both the executable and a folder containing several JAR-files supplied with openBIS are required. Login credentials for openBIS access can be specified in the following section (`openBIS`) and the final keyword group holds the path to the local source of this package. It is only used to update the databases in `/data` (see section B.3) and therefore will not be needed in production environments, unless the metadata that comes with singleCellData is outdated or this package is used for data not produced by InfectX.

B.2 Short Package Demonstration

This short demonstration of singleCellFeatures may serve as an entry point for readers interested in using the package for accessing and processing InfectX data. More code examples are available as vignettes.

```
library(singleCellFeatures)
## for Rhino, some metadata might be incomplete for plates:
##   R10-40: 183 wells
## for Salmonella, some metadata might be incomplete for plates:
##   VZ018-1C: 104 wells
##   VZ019-1C: 104 wells
##   VZ020-1C: 104 wells
##   VZ021-1C: 104 wells
##   VZ022-1C: 95 wells
##   VZ023-1C: 104 wells
##   VZ024-1C: 104 wells
##   VZ025-1C: 105 wells
##   VZ026-1C: 105 wells
##   VZ027-1C: 104 wells
## for Uukuniemi, no well database found
## missing metadata for 191 plates.
## coverage: 0.95410860163383
## run wellDatabaseCoverage(TRUE) to show all missing plates.
```

Whenever loading the package, the `.onLoad` hook triggers the generation of a short report on metadata coverage. This essentially checks that detailed metadata information is present in metadata databases (see section B.3.1) for all plates on openBIS and reports on any unavailable data. Issuing the command

`wellDatabaseCoverage(TRUE)` will show a more detailed report, explicitly listing all barcodes of missing plates.

```
wells  <- findWells(experiment="brucella-du-k[12]", content="MTOR")
## there are 8 wells remaining:
##   J101-2C  H6   SIRNA  DHARMACON_L-003008-00_A  2475  MTOR
##   J107-2D  H6   SIRNA  DHARMACON_L-003008-00_C  2475  MTOR
##   J110-2D  H6   SIRNA  DHARMACON_L-003008-00_D  2475  MTOR
##   J104-2C  H6   SIRNA  DHARMACON_L-003008-00_B  2475  MTOR
##   J107-2C  H6   SIRNA  DHARMACON_L-003008-00_C  2475  MTOR
##   J110-2C  H6   SIRNA  DHARMACON_L-003008-00_D  2475  MTOR
##   J101-2D  H6   SIRNA  DHARMACON_L-003008-00_A  2475  MTOR
##   J104-2D  H6   SIRNA  DHARMACON_L-003008-00_B  2475  MTOR
plates <- lapply(wells, convertToPlateLocation)
```

The regular expression `brucella-du-k[12]` will restrict the search to either the K1 or K2 screens of Dharmacon unpooled *Brucella* plates and within this set all wells are selected that contain MTOR targeting siRNA (for further parameters, see table B.1). The resulting `WellLocation` objects can be converted to `PlateLocation` structures for fetching complete plates instead of single wells.

```
data <- PlateData(plates[[1]])
## reading plate chache file.
## assuming 9 images per well:
## max legnth: 3456, fraction of full length features: 0.995
## removing 3 features (length == 1):
##   Bacteria.SubObjectFlag
##   Batch_handles
##   Image.ModuleError_43CreateBatchFiles
data <- extractFeatures(data,
                        select=c("^Cells.", "^Nuclei.", "^PeriNuclei.",
                                 "^VoronoiCells."),
                        drop=c("CorrPathogen", "Bacteria",
                               "_MedianUpperTwoPercentIntensity_",
                               "_MedianUpperFivePercentIntensity_",
                               "_MedianUpperTenPercentIntensity_"))
## removing 84 unmatched features.
## removing 190 matched features.
```

Loading of a complete `PlateData` object will issue several sanity checks to ensure the resulting dataset is consistent throughout wells and all expected features are available, while removing spurious features. In order to detect

whether there are 6 or 9 images per well, a heuristic determines the fraction of features which contain as many slots as are maximally encountered. As not all features are of interest, a subset can be extracted, based on the two (vectors of) regular expressions supplied as `select` and `drop` arguments.

```
data <- augmentImageLocation(data)
data <- augmentCordinateFeatures(data, ellipse=1, facet=NULL,
                                 center.dist=FALSE, density=FALSE)
## using a single ellipse, 100px (within images) and
## 200px (within wells) dist from borders.
```

Next, several functions are used to augment the dataset with new features, the first two of which are concerned with geometric data such as image location within well and object location within image. Here, all location features are expanded to include object location within well (by using the information of image location within well and location information within image) and object locations are categorized with respect to two ellipses, one at well level and the other at image level.

```
data <- augmentBscore(data, features=c(".AreaShape_", ".Intensity_",
                                       ".Texture_"),
                      func.aggr="mean")
data <- augmentMars(data, bscore=TRUE,
                    model=c("^Nuclei.Location_In_Ellipse_Well$",
                            "^Nuclei.Location_In_Ellipse_Image$"),
                    features=c(".AreaShape_", ".Intensity_",
                               ".Texture_"))
## dropping 1 features due to zero variance:
##   Nuclei.AreaShape_EulerNumber
## normalizing 295 features,
## model terms include bscoring and:
##   Nuclei.Location_In_Ellipse_Well
##   Nuclei.Location_In_Ellipse_Image
```

Augmenting data with B-scores will calculate row-, column- and plate-effects for each of the matched features. This is followed by MARS modeling, which, in the present case, estimates the effect of the corresponding B-scores and the previously generated ellipse features on each of the selected features, returning only residuals. Further terms may be included, such as object densities or nucleus size, using the `model` argument (see section 5.3.3).

```
data <- augmentAggregate(data, features="_MARSed$", level="plate",
                         func.aggr="median")
data <- augmentAggregate(data, features="_MARSed$", level="well",
                         neighbors=TRUE,  func.aggr="mad")
data <- normalizeData(data, values="MARSed$",
                      center="MARSed_Aggreg_P_median$",
                      scale="MARSed_Aggreg_N_mad$")
```

The `augmentAggregate` function is used to generate aggregated values at well of plate level, which is needed for centering and scaling feature data. First, for each of the MARS residuals, the plate median is calculated, followed by MAD values at well level. To make this step more robust, the option of including 4 neighbors of each well (left, right, top and bottom) is enabled. Using the intermediate features, a call to `normalizeData` will produce normalized features. Here all variables resulting from MARS analysis are centered using plate medians and scaled with corresponding MAD values per well.

```
data <- meltData(data)
data <- moveFeatures(data, to="Cells",
                     from=c("_Normed", "Well.Type", "Well.Gene_Name"))
```

Finally, the `PlateData` object is molten into a list of `data.frames`. In order to obtain a single `data.frame`, containing all data of interest, features can be moved between nodes of the molten data structure by using the `moveFeatures` function. Cases where dimensions disagree (for example when moving a feature that is scalar valued per well, e.g. well index, to a node containing features that are vector valued per image) are handled automatically by duplicating or aggregating data.

This small example is intended highlight some of the flexibility that comes with singleCellFeatures. Due to there not being a clear-cut analysis procedure that can be applied to all datasets, the package is designed to accommodate future applications as good as possible at the expense of some complexity.

B.3 Metadata Databases

Metadata files are used in two separate contexts, the first being generation of lookup tables for dataset searches and feature availability checks and the second is involved with compilation of relevant metadata upon import of new data (see section 4.2). Three different types of CSV-based files are required, a

plate database, feature lists and aggregate files, creation and storage of which is described in the following paragraphs.

For production use of singleCellFeatures, only aggregate files are required, as the other two file types are available in processed form as package data files, distributed with the package. All the metadata contained in aggregate files could unfortunately not be supplied directly with the package due to large size and more importantly, because they hold information that currently may not be released to the public as a whole (e.g. sequences of all siRNAs used throughout all screens, only a subset of which is currently published).

B.3.1 Plate Database

The purpose of this file is creating a table of all plates that have associated single cell feature data available. This information is used for assessing the coverage of actual metadata and for lookup of data location both locally and on openBIS. Knowing how much of the data is represented in metadata files is important, as only that subset is available to singleCellFeatures. Upon loading the package in an R session, the extent of coverage is surveyed and reported in order to warn the user when, for example, outdated metadata is used that only contains an older set of available data.

Creation. In openBIS, choose `Browse > Data Set Search`, select the drop-down option `Data Set Type` and put in `cc` as keyword for searching for all single cell feature datasets. Make sure all available columns are displayed by selecting `Settings` and checking all `Visible?` boxes. Finally choose `Export` to save the table to disk.

Names and content. The file is expected to be named as `HCS_ANALYSIS_CELL_FEATURES_CC_MAT.tsv` and be located directly at the path specified as `metaDir`. The corresponding database for the R package is located in the package `/data` directory, is saved as `/plateDatabase.rda` and the object that is attached when calling `data(plateDatabase)` in R is named `plate.database`. The table contains columns `Barcode`, `Space`, `Group`, `Experiment` and `DataID`, which for each barcode, defines the location of associated single cell feature datasets within openBIS and the local cache hierarchy, which mirrors the structure on openBIS.

B.3.2 Feature Lists

For most of the time during development of singleCellFeatures, work on storage infrastructure at the University of Basel introduced issues when download-

ing datasets that could lead to features missing from the downloaded data without the user being warned about this. A simple fix is provided in the form of a feature list database that specifies a set of features that are expected to be present for each pathogen. This approach will report both false positives (features that are expected but not actually available for the given dataset) and false negatives (features that are not expected but still available), as the set of available features not only depends on pathogen, but also on the state of the analysis pipeline used for feature extraction.

A more sophisticated solution involves querying the metadata database of openBIS for all feature files per plate and the increased granularity would consequently eliminate false reports. Such an approach was contemplated but as of yet could not be implemented due to time constraints.

Creation. In openBIS, for each project (e.g. `ADENO_TEAM`), display all experiments, choose the most recent (which seems like a regular screen, e.g. `ADENO-AU-CV2`), choose any plate (all are assumed to contain the same set of features) and list all available data sets sorted by data set type. Pick the most recent data set of type `HCS_ANALYSIS_CELL_FEATURES_CC_MAT` and list all files belonging to that data set. This view is then copy-pasted into a text editor and all files that do not end in `.mat` are removed by hand (usually only 1–3 files).

Names and content. The raw data files used for updating the database associated with singleCellFeatures are expected to be located in a subdirectory to `metaDir` and saved as `PATHOGEN-*.txt` (for example `ADENO-AU-CV2.txt`). The pathogen name should be capitalized and specification of the screen that was used for obtaining the information may follow but is not required. Each line contains the name of a feature that will be expected to be present in all screens of the given pathogen, followed by a space, separating the name from further information which will be ignored. The resulting R data file is named `featureDatabase.rda`, the object name is `feature.database` and represents the data as a list of character vectors with slots for pathogens.

B.3.3 Aggregate Files

This set of files is most important, as it is required for correct operation of singleCellFeatures, whereas others are only needed for updating the corresponding data files distributed with the package itself. All metadata that is used, both for searching for datasets from within singleCellFeatures, as well as for creating `MetaData` objects, originates from aggregate files.

Creation. Many different types of aggregate files are available from the InfectX openBIS instance, compiled with different aggregation procedures and only the recent most generation contains all required columns. For accessing the files, in openBIS, choose the `Data Sets` tab under `INFECTX/_COMMOM/REPORTS`) and sort by code. The files produced by the current aggregation procedure (imported into openBIS at the end of May 2015) are the best choice and column naming expectations of singleCellFeatures are based on this generation of aggregates (examples include *Brucella* and *Salmonella*).

Names and content. Aggregate files are expected to reside in a subdirectory under `metaDir`, called `Aggregates` and follow the naming pattern `Pathogen Report_*.csv`, where the pathogen name is specified in camel case and the specification of openBIS document ID following an underline is optional (e.g. `AdenoReport_20150522-0936.csv`). These files are large (up to 200 MB) and constitute of rows representing wells corresponding to screens of a given pathogen and 55 columns, of which `Barcode`, `BATCH`, `eCount_oCells`, `Experiment`, `GENESET`, `Group`, `ID`, `ID_openBIS`, `LIBRARY`, `Name`, `PATHOGEN`, `PLATE_QUALITY_STATUS`, `PLATE_TYPE`, `REPLICATE`, `Seed_sequence_antisense_5_3`, `Sequence_antisense_5_3`, `Sequence_target_sense_5_3`, `Space`, `WELL_QUALITY_STATUS`, `WellColumn`, `WellRow` and `WellType` are relevant to singeCellFeatures. Please refer to section 4.1.1 for more information on the individual columns.

The corresponding lookup tables store the columns `Barcode`, `WellRow`, `Well Column`, `WellType`, `ID_openBIS`, `ID` and `Name` as `wellDatabasePathogen.rda` where the pathogen name is spelled in camel case (for example `wellDatabase Adeno.rda`) and the data frame in R is called `well.database.pathogen` where the pathogen name is in lowercase letters (e.g. `well.database.adeno`). The primary purpose of well-level metadata lookup tables is fast searching, as constant loading of large pathogen-level aggregate files is time consuming. A further type of aggregate-derived files are plate metadata caches. These are saved alongside the directory structure that organizes downloaded data and their naming scheme follows `barcode_metadata.rds` (e.g. `J101-2C_metadata.rds`). Plate-level metadata caches consist of all columns available in aggregate files and the 384 rows corresponding to the given plate and again constitute an optimization mechanism for reducing the time required for serving complete `Data` objects.

B.4 Dataset search

Two functions for searching datasets are available, `findPlates` and `findWells`, both of which return lists of respective `DataLocation` objects. The arguments common to both functions along are `pathogens`, `experiments`, `plates`, `well.`

Table B.1: Two functions for identifying datasets of interest are available, `findPlates` and `findWells`. The first set of parameters is available to both functions, while the three arguments separated at the bottom can only be specified in `findWells`. The column specifying targeted metadata fields is referring to names explained in tables 4.1 and 4.2.

Parameter name	Metadata column	Description
`pathogens = NULL`	`experiment.group`	Vector of strings, case insensitive, matched by adding the suffix `_TEAM` to the input.
`experiments = NULL`	`experiment.name`	Vector of strings, applied as individual case insensitive regular expressions.
`plates = NULL`	`plate.barcode`	Vector of strings, case sensitive, matched exactly.
`well.types = NULL`	`well.type`	Vector of strings, applied as individual case insensitive regular expressions.
`contents = NULL`	`sirna.name`, `gene.id` or `gene.name`	Vector of strings, case insensitive, each string is matched against any of the three columns.
`id.openBIS = NULL`	`sirna.name`	Vector of strings, applied as individual case insensitive regular expressions.
`id.infx = NULL`	`gene.id`	Vector of strings, applied as individual case insensitive regular expressions.
`name = NULL`	`gene.name`	Vector of strings, applied as individual case insensitive regular expressions.
`verbose = FALSE`	–	Logical value indicating whether to produce more verbose output.
`well.rows = NULL`	`well.row`	Vector of single characters ($\in \{A, B, \ldots, P\} \wedge \{a, b, \ldots, p\}$), matched exactly.
`well.cols = NULL`	`well.col`	Vector of single integers ($\in \{1, 2, \ldots, 24\}$), matched exactly.
`well.names = NULL`	`well.row` and `well.col`	Vector of strings ($\in \{A1, A2, \ldots, A24, B1, \ldots, P24\}$), matched exactly.

`types`, `contents`, `id.openBIS`, `id.infx`, `name` and `verbose`. All, except for the last parameter, which is a logical value for increased verbosity, can accept vector valued arguments that are matched against corresponding metadata columns. Please refer to table B.1 for more details on how values are matched and which metadata columns (see tables 4.1 and 4.2) are involved.

In order to achieve quick lookups, not the metadata files themselves are subject to search, but rather an indexed version in the form of a plate database (see section B.3.1) and pathogen specific well databases (see section B.3.3). This reduces the time frame required for performing searches considerably, especially in pathogen spanning cases, as the relevant data files that have to be loaded are 1–3 MB, which corresponds to a 100-fold reduction in size, and therefore can be attached instantly.

Epilogue

Building on experiences gained from investigations into InfextX single cell feature data such as outlined in sections 5.2 and 5.3.3, Drewek (2015) suggests using the random forest algorithm (Breiman 2001; Liaw and Wiener 2002) for determining which features are most influential for infection response. Random forest is chosen due to its robustness towards over-fitting and good performance under complex interactions, which both are important properties given the high dimensional, heavily correlated datasets at hand. Furthermore, an elegant estimation of variable importance is available by random forest analysis.

Predicting the results from DTIS using all cellular features measured on the DNA and actin channels yields an importance score assigned to each of the included features. By generating rankings per well and not aggregating data in well replicates, normalization can be reduced to mean centering. An overall importance score can be calculated in a subsequent step by averaging the respective importance scores from replicated experiments. This may alleviate some of the problems surrounding data normalization encountered in earlier analysis approaches.

A visualization of results obtained via the described approach is shown in figure E.1. Similarity of overall importance scores for the ~300 included cellular features is described by complete linkage hierarchical clustering based on euclidean distances. Interestingly, the two included down-hits from PMM analysis (see section 5.1.2), MTOR (H6) and TGFBR1 (M4), are grouped closely together, as are the two selected up-hits, PIK3R3 (K8) and RIPK4 (G17). Moreover, 3 of the 4 included control wells form their own cluster (H2, G23 and L1), with the exception of scrambled well J2. Also, the well location with respect to assay plate does not seem to matter much given these results. Of course this provides only little evidence for the aptitude of random forest but may never-

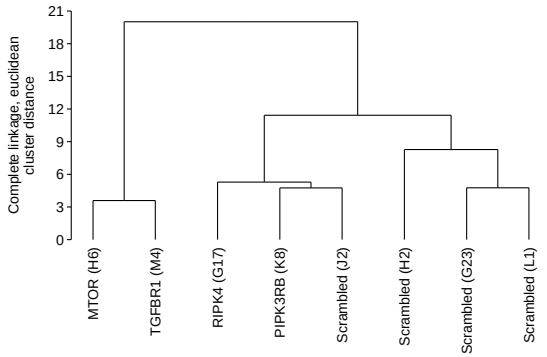


Figure E.1: Complete linkage hierarchical clustering performed on euclidean distances from cell feature importance scores, obtained through random forest analysis and averaged by well. For each gene and scrambled well, all available 8 replicates of the *Brucella* Dharmacon unpooled dataset were considered.

theless present a promising motive to pursue this avenue further. One should certainly study a larger range of genes and it would be interesting to see to what extent known gene networks can be recovered using these methods.

One additional possibility, which so far has not been explored and may be of value to random forest analysis, is utilizing replicates and the behavior of control wells to correct for experimental and biological noise. Looking at heatmap representations of correlation matrices (see figure E.2) based on importance scores (Pearson's correlation; red) and feature rankings (Spearman's rank correlation; blue) per individual well, two observations can be made: *(1)* there is considerable variation among replicates and *(2)* scrambled wells behave similarly to targeted siRNA wells, with respect to this comparison.

Exploiting the large number of replicates available in kinome-wide screens is an opportunity not available to the analysis of genome-wide datasets which were investigated by Drewek. Both stability of rankings and variability of importance scores may be investigated and this could be used to remove outliers. Additionally, a more sophisticated scheme for aggregating the scores from individual wells should be developed. The current averaging can easily be improved upon, for example by giving more weight to patterns common to several wells, thus reducing the influence of deviating wells. Furthermore, aggregation of rankings within gene targets, but spanning manufacturers and possibly even pathogens could be attempted.

Observation *(2)* is mentioned because the initial presumption that the two well types somehow behave differently, perhaps with scrambled wells resulting in

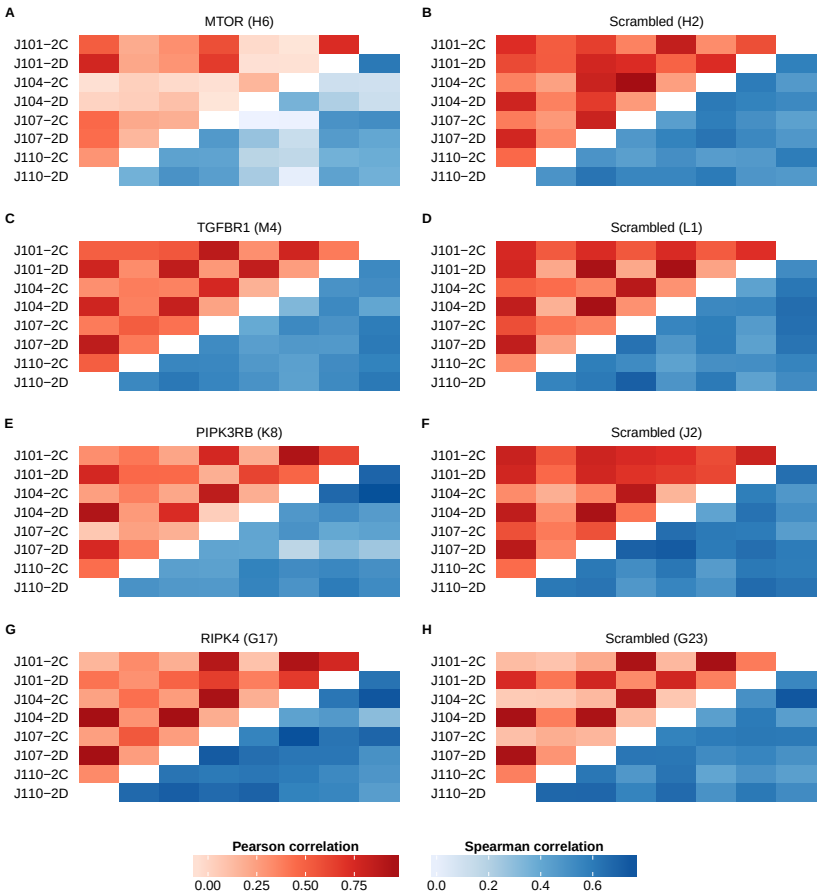


Figure E.2: Heatmap representations of correlation matrices obtained by comparing importance scores of features as determined by random forest analysis. Red shades indicate Pearson's product-moment correlation while blue shades visualize Spearman's rank correlation coefficients. For each gene and scrambled well, all available 8 replicates of the *Brucella* Dharmacon unpooled dataset were considered.

lower correlation coefficients due to less clear patterns, cannot be confirmed based on these results. This is not neccessarily an issue of data quality but may simply be a siRNA-induced phenotype. It does, however, raise interesting questions: Do scrambled wells manifest some type of 'background behavior' of non-specific response to siRNA reagents? How could this source of biological noise be summarized and subtracted from targeted siRNA wells? How do mock control wells behave and could they be exploited as well?

Already from this brief look at new developments in analysis of InfectX single cell feature data, it becomes apparent that much opportunity is present within this invaluable dataset. Further study should be able to bring to light exciting discoveries in the study of the human infectome and likely enable new ways of analyzing image-based HTS data at single cell level.

Bibliography

Alberts, Bruce, Alexander Johnson, Julian H. Lewis, Martin Raff, Keith Roberts, and Peter Walter. 2008. *Molecular Biology of the Cell.* 5th ed. New York: Garland Science.

Allen, Michael Patrick. 1997. *Understanding Regression Analysis.* New York: Plenum Press. doi:10/fc84wf.

Anderson, Burt E., and Mark A. Neuman. 1997. "*Bartonella* spp. as Emerging Human Pathogens." *Clinical Microbiology Reviews* 10 (2): 203–219.

Atluri, Vidya L., Mariana N. Xavier, Maarten F. de Jong, Andreas B. den Hartigh, and Renée M. Tsolis. 2011. "Interactions of the Human Pathogenic *Brucella* Species with Their Hosts." *Annual Review of Microbiology* 65 (1): 523–541. doi:10/bhsrnd.

Baltimore, David. 1971. "Expression of Animal Virus Genomes." *Bacteriological Reviews* 35 (3): 235–241.

Bargen, Kristine von, Jean-Pierre Gorvel, and Suzana P. Salcedo. 2012. "Internal Affairs: Investigating the *Brucella* Intracellular Lifestyle." *FEMS Microbiology Reviews* 36 (3): 533–562. doi:10/bb24.

Barnes, Allan M. 1990. "Plague in the U.S.: Present and Future." In *Proceedings of the Fourteenth Vertebrate Pest Conference,* 43–46. Lincoln: University of Nebraska.

Barsy, Marie de, Alexandre Jamet, Didier Filopon, Cécile Nicolas, Géraldine Laloux, Jean-François Rual, Alexandre Muller, et al. 2011. "Identification of a *Brucella* spp. Secreted Effector Specifically Interacting with Human Small GTPase Rab2." *Cellular Microbiology* 13 (7): 1044–1058. doi:10/b8s3r2.

Bates, Douglas, and Martin Maechler. 2015. *Matrix: Sparse and Dense Matrix Classes and Methods.* Version 1.2-2. http://matrix.r-forge.r-project.org.

Bauch, Angela, Izabela Adamczyk, Piotr Buczek, Franz-Josef Elmer, Kaloyan Enimanev, Pawel Glyzewski, Manuel Kohler, et al. 2011. "OpenBIS: A flexible Framework for Managing and Analyzing Complex Data in Biology Research." *BMC Bioinformatics* 12 (1): 468. doi:10/c6pjh8.

Bengtsson, Henrik, Andy Jacobson, and Jason Riedy. 2015. *R.matlab: Read and Write MAT Files and Call MATLAB from Within R.* Version 3.2.0. https://github.com/HenrikBengtsson/R.matlab.

Bhinder, Bhavneet, and Hakim Djaballah. 2013. "Systematic Analysis of RNAi Reports Identifies Dismal Commonality at Gene-Level and Reveals an Unprecedented Enrichment in Pooled shRNA Screens." *Combinatorial Chemistry & High Throughput Screening* 16 (9): 665–681. doi:10/bb25.

Bhinder, Bhavneet, David Shum, and Hakim Djaballah. 2014. "Comparative Analysis of RNAi Screening Technologies at Genome-Scale Reveals an Inherent Processing Inefficiency of the Plasmid-Based shRNA Hairpin." *Combinatorial Chemistry & High Throughput Screening* 17 (2): 98–113. doi:10/bb26.

Boutros, Michael, Lígia P. Brás, and Wolfgang Huber. 2006. "Analysis of Cell-Based RNAi Screens." *Genome Biology* 7 (7): R66.1–R66.11. doi:10/b5n8zb.

Bowman, Adrian W., and Adelchi Azzalini. 1997. *Applied Smoothing Techniques for Data Analysis: The Kernel Approach with S-Plus Illustrations.* Oxford: Oxford University Press.

Breiman, Leo. 2001. "Random Forests." *Machine Learning* 45 (1): 5–32. doi:10/d8zjwq.

Brown, Bruce M. 2006. "Tukey's Median Polish." In *Encyclopedia of Statistical Sciences,* 2nd ed., edited by Samuel Kotz, Campbell B. Read, Narayanaswamy Balakrishnan, and Brani Vidakovic. New York: John Wiley & Sons, Inc. doi:10/dcrfhh.

Carpenter, Anne E., Thouis R. Jones, Michael R. Lamprecht, Colin Clarke, In Han Kang, Ola Friman, David A. Guertin, et al. 2006. "CellProfiler: Image Analysis Software for Identifying and Quantifying Cell Phenotypes." *Genome Biology* 7 (10): R100.1–R100.11. doi:10/fqfn3j.

Carthew, Richard W., and Erik J. Sontheimer. 2009. "Origins and Mechanisms of miRNAs and siRNAs." *Cell* 136 (4): 642–655. doi:10/brk2cm.

Celli, Jean, Suzana P. Salcedo, and Jean-Pierre Gorvel. 2005. "*Brucella* Coopts the Small GTPase Sar1 for Intracellular Replication." *PNAS* 102 (5): 1673–8. doi:10/fqjz5r.

Cerutti, Heriberto, and Armando J. Casas-Mollano. 2006. "On the Origin and Functions of RNA-Mediated Silencing: From Protists to Man." *Current Genetics* 50 (2): 81–99. doi:10/fm4n28.

Condit, Richard C., Nissin Moussatche, and Paula Traktman. 2006. "In a Nutshell: Structure and Assembly of the Vaccinia Virion." *Advances in Virus Research* 66:31–124. doi:10/cqhj4d.

Cossart, Pascale, and Alice Lebreton. 2014. "A Trip in the 'New Microbiology' with the Bacterial Pathogen *Listeria monocytogenes*." *Federation of European Biochemical Societies Letters* 588 (15): 2437–45. doi:10/bb29.

Cossart, Pascale, and Philippe J. Sansonetti. 2004. "Bacterial Invasion: The Paradigms of Enteroinvasive Pathogens." *Science* 304 (5668): 242–248. doi:10/b958vb.

Craighead, John E. 2000. *Pathology and Pathogenesis of Human Viral Disease*. San Diego: Academic Press. doi:10/cv2w6k.

Croxen, Matthew A., and Brett B. Finlay. 2010. "Molecular Mechanisms of *Escherichia coli* Pathogenicity." *Nature Reviews Microbiology* 8 (1): 26–38. doi:10/fmbv4h.

Czyż, Daniel M., Lakshmi-Prasad Potluri, Neeta Jain-Gupta, Sean P. Riley, Juan J. Martinez, Theodore L. Steck, Sean Crosson, Howard A. Shuman, and Joëlle E. Gabay. 2014. "Host-Directed Antimicrobial Drugs with Broad-Spectrum Efficacy Against Intracellular Bacterial Pathogens." *mBio* 5 (4): e01534–14. doi:10/bb3b.

Dehio, Christoph. 2005. "*Bartonella*-Host-Cell Interactions and Vascular Tumour Formation." *Nature Reviews Microbiology* 3 (8): 621–631. doi:10/bbtb3s.

Dehio, Christoph, Marlene Meyer, Jürgen Berger, Heinz Schwarz, and Christa Lanz. 1997. "Interaction of *Bartonella henselae* with Endothelial Cells Results in Bacterial Aggregation on the Cell Surface and the Subsequent Engulfment and Internalisation of the Bacterial Aggregate by a Unique Structure, the Invasome." *Journal of Cell Science* 110 (18): 2141–2154.

Dehio, Michaela, Alexander Knorre, Christa Lanz, and Christoph Dehio. 1998. "Construction of Versatile High-Level Expression Vectors for *Bartonella henselae* and the Use of Green Fluorescent Protein as a New Expression Marker." *Gene* 215 (2): 223–229. doi:10/c6z34k.

Dobson, Andrew P., and Robin E. Carper. 1996. "Infectious Diseases and Human Population History." *BioScience* 46 (2): 115–126. doi:10/fncwmm.

Drewek, Anna. 2015. "Statistical Inference on Pathogen Entry Into Human Cells." PhD diss., ETH Zürich.

Durinck, Steffen, Yves Moreau, Arek Kasprzyk, Sean Davis, Bart De Moor, Alvis Brazma, and Wolfgang Huber. 2005. "BioMart and Bioconductor: a powerful link between biological databases and microarray data analysis." *Bioinformatics* 21 (16): 3439–3440. doi:10/c9b4zt.

Durinck, Steffen, Paul T. Spellman, Ewan Birney, and Wolfgang Huber. 2009. "Mapping Identifiers for the Integration of Genomic Datasets with the R/Bioconductor package biomaRt." *Nature Protocols* 4 (8): 1184–1191. doi:10/c4b7dd.

Echeverri, Christophe J., and Norbert Perrimon. 2006. "High-Throughput RNAi Screening in Cultured Cells: A User's Guide." *Nature Reviews Genetics* 7 (5): 373–384. doi:10/fmrrcj.

Fàbrega, Anna, and Jordi Vila. 2013. "*Salmonella enterica* Serovar Typhimurium Skills to Succeed in the Host: Virulence and Regulation." *Clinical Microbiology Reviews* 26 (2): 308–341. doi:10/bb3c.

Farber, Jeffrey M., and Pearl I. Peterkin. 1991. "*Listeria monocytogenes*, a Food-Borne Pathogen." *Microbiological Reviews* 55 (3): 476–511.

Friedman, Jerome H. 1991. "Multivariate Adaptive Regression Splines." *The Annals of Statistics* 19 (1): 1–67. doi:10/ctgh88.

Friedman, Jerome, Trevor Hastie, and Rob Tibshirani. 2010. "Regularization Paths for Generalized Linear Models via Coordinate Descent." *Journal of Statistical Software* 33 (1): 1–22. doi:10/bb3d.

Gábor, Dennis. 1946. "Theory of Communication." *The Journal of the Institution of Electrical Engineers - Part III: Radio and Communication* 93 (26): 429–441. doi:10/2rz.

Geier, Florian, Matthias Truttmann, and Christoph Dehio. 2010. *Mathematical Modeling of Infection Pathways.* Presentation, InfectX Retreat, Leuenberg, June 3–4.

Gelman, Andrew, and Jennifer Hill. 2007. *Data Analysis Using Regression and Multilevel/Hierarchical Models.* New York: Cambridge University Press. doi:10/dbrqk6.

Gillespie, Stephen H., and Peter M. Hawkey. 2006. *Principles and Practice of Clinical Bacteriology.* 2n ed. 295–304. West Sussex: John Wiley & Sons, Ltd. doi:10/c6b8p2.

Gray, Kristian A., Louise C. Daugherty, Susan M. Gordon, Ruth L. Seal, Mathew W. Wright, and Elspeth A. Bruford. 2013. "Genenames.org: The HGNC Resources in 2013." *Nucleic Acids Research* 41 (Database issue): D545–D552. doi:10/bb3h.

Haglund, Cat M., and Matthew D. Welch. 2011. "Pathogens and Polymers: Microbe–Host Interactions Illuminate the Cytoskeleton." *The Journal of Cell Biology* 195 (1): 7–17. doi:10/d99p88.

Ham, Hyeilin, Anju Sreelatha, and Kim Orth. 2011. "Manipulation of Host Membranes by Bacterial Effectors." *Nature Reviews Microbiology* 9 (9): 635–646. doi:10/cvbpgm.

Hannon, Gregory J. 2002. "RNA Interference." *Nature* 418 (6894): 244–251. doi:10/frt8fp.

Haraga, Andrea, Maikke B. Ohlson, and Samuel I. Miller. 2008. "Salmonellae Interplay with Host Cells." *Nature Reviews Microbiology* 6 (1): 53–66. doi:10/cf4h5k.

Haralick, Robert M., Karthikeyan Shanmugam, and Its'Hak Dinstein. 1973. "Textural Features for Image Classification." *IEEE Transactions on Systems, Man and Cybernetics* SMC-3 (6): 610–621. doi:10/bdqvtn.

Harms, Alexander, and Christoph Dehio. 2012. "Intruders Below the Radar: Molecular Pathogenesis of *Bartonella* spp." *Clinical Microbiology Reviews* 25 (1): 42–78. doi:10/fzwksz.

Harrison, Stephen C. 2008. "Viral Membrane Fusion." *Nature Structural & Molecular Biology* 15 (7): 690–698. doi:10/cmtmj5.

Hastie, Trevor, and Stephen Milborrow. 2015. *Earth: Multivariate Adaptive Regression Splines.* Version 4.4.2. https://cran.r-project.org/web/packages/earth.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning.* 2nd ed. New York: Springer. doi:10/cd7nhz.

Hawn, Thomas R., Javeed A. Shah, and Daniel Kalman. 2015. "New Tricks for Old Dogs: Countering Antibiotic Resistance in Tuberculosis with Host-Directed Therapeutics." *Immunological Reviews* 264 (1): 344–362. doi:10/bb3w.

Hayashi, Fumio. 2000. *Econometrics.* Princeton: Princeton University Press.

Honeychurch, Kady M., Guang Yang, Robert Jordan, and Dennis E. Hruby. 2007. "The Vaccinia Virus F13L YPPL Motif is Required for Efficient Release of Extracellular Enveloped Virus." *Journal of Virology* 81 (13): 7310–7315. doi:10/ft946w.

Huang, Ju, and John H. Brumell. 2014. "Bacteria-Autophagy Interplay: A Battle for Survival." *Nature Reviews Microbiology* 12 (2): 101–114. doi:10/bb33.

Huber, Wolfgang, Vincent J. Carey, Robert Gentleman, Simon Anders, Marc Carlson, Benilton S. Carvalho, Hector C. Bravo, et al. 2015. "Orchestrating High-Throughput Genomic Analysis With Bioconductor." *Nature Methods* 12 (2): 115–121. doi:10/bb35.

Hulo, Chantal, Edouard de Castro, Patrick Masson, Lydie Bougueleret, Amos Bairoch, Ioannis Xenarios, and Philippe Le Mercier. 2011. "ViralZone: A Knowledge Resource to Understand Virus Diversity." *Nucleic Acids Research* 39 (Database issue): D576–D582. doi:10/cj4gw6.

Jacobs, Samantha E., Daryl M. Lamson, Kirsten St George, and Thomas J. Walsh. 2013. "Human Rhinoviruses." *Clinical Microbiology Reviews* 26 (1): 135–162. doi:10/bb36.

Johnson, Sam A., and Tony Hunter. 2005. "Kinomics: Methods for Deciphering the Kinome." *Nature Methods* 2 (1): 17–25. doi:10/bx6n3z.

Jurgeit, Andreas, Robert McDowell, Stefan Moese, Eric Meldrum, Reto Schwendener, and Urs F. Greber. 2012. "Niclosamide Is a Proton Carrier and Targets Acidic Endosomes with Broad Antiviral Effects." *PLOS Pathogens* 8 (10): e1002976. doi:10/bb37.

Jurgeit, Andreas, Stefan Moese, Pascal Roulin, Alexander Dorsch, Mark Lötzerich, Wai-Ming Lee, and Urs F. Greber. 2010. "An RNA Replication-Center Assay for High Content Image-Based Quantifications of Human Rhinovirus and Coxsackievirus Infections." *Virology Journal* 7 (1): 264.1–264. doi:10/ddm3ss.

Kamentsky, Lee, Thouis R. Jones, Adam Fraser, Mark-Anthony Bray, David J. Logan, Katherine L. Madden, Vebjorn Ljosa, Curtis Rueden, Kevin W. Eliceiri, and Anne E. Carpenter. 2011. "Improved Structure, Function and Compatibility for CellProfiler: Modular High-Throughput Image Analysis Software." *Bioinformatics* 27 (8): 1179–1180. doi:10/bmqwk2.

Kanehisa, Minoru, and Susumu Goto. 2000. "KEGG: Kyoto Encyclopedia of Genes and Genomes." *Nucleic Acids Research* 28 (1): 27–30. doi:10/b9st54.

Kasper, Christoph A. 2012. "Amplifying the Innate Immune Response: Cell-Cell Propagation of Proinflammatory Signals During Bacterial Infection." PhD diss., University of Basel. doi:10/bb4j.

Kim, Daniel H., and John J. Rossi. 2007. "Strategies for Silencing Human Disease Using RNA Interference." *Nature Reviews Genetics* 8 (3): 173–184. doi:10/fvp44b.

Knapp, Bettina, Ilka Rebhan, Anil Kumar, Petr Matula, Narsis A. Kiani, Marco Binder, Holger Erfle, et al. 2011. "Normalizing for Individual Cell Population Context in the Analysis of High-Content Cellular Screens." *BMC Bioinformatics* 12 (1): 485.1–485.14. doi:10/fzr79d.

Konis, Kjell Peter. 2007. "Linear Programming Algorithms for Detecting Separated Data in Binary Logistic Regression Models." PhD diss., University of Oxford. http://ora.ox.ac.uk/objects/ora:2848.

Kopecko, Dennis J., Lan Hu, and Kristien J.M. Zaal. 2001. "*Campylobacter jejuni* – Microtubule-Dependent Invasion." *Trends in Microbiology* 9 (8): 389–396. doi:10/bw9rq9.

Kosmidis, Ioannis. 2007. "Bias Reduction in Exponential Family Nonlinear Models." PhD diss., University of Warwick. http://ethos.bl.uk/OrderDetails.do?uin=uk.bl.ethos.492241.

Leirião, Patrícia, Cristina D. Rodrigues, Sónia S. Albuquerque, and Maria M. Mota. 2004. "Survival of Protozoan Intracellular Parasites in Host Cells." *EMBO Reports* 5 (12): 1142–1147. doi:10/csc3ph.

Lenaerts, Liesbeth, Erik De Clercq, and Lieve Naesens. 2008. "Clinical Features and Treatment of Adenovirus Infections." *Reviews in Medical Virology* 18 (6): 357–374. doi:10/cnh3zx.

Liaw, Andy, and Matthew Wiener. 2002. "Classification and Regression by randomForest." *R News* 2 (3): 18–22.

Littman, Robert J. 2009. "The Plague of Athens: Epidemiology and Paleopathology." *Mount Sinai Journal of Medicine* 76 (5): 456–467. doi:10/bztbrj.

Maglott, Donna, Jim Ostell, Kim D. Pruitt, and Tatiana Tatusova. 2011. "Entrez Gene: Gene-Centered Information at NCBI." *Nucleic Acids Research* 39 (Database issue): D52–D57. doi:10/fsjcqz.

Majowicz, Shannon E., Jennie Musto, Elaine Scallan, Frederick J. Angulo, Martyn Kirk, Sarah J. O'Brien, Timothy F. Jones, Aamir Fazil, and Robert M. Hoekstra. 2010. "The Global Burden of Nontyphoidal *Salmonella* Gastroenteritis." *Clinical Infectious Diseases* 50 (6): 882–889. doi:10/b33mj3.

Malo, Nathalie, James A Hanley, Sonia Cerquozzi, Jerry Pelletier, and Robert Nadon. 2006. "Statistical Practice in High-Throughput Screening Data Analysis." *Nature Biotechnology* 24 (2): 167–175. doi:10/dqc8ck.

Manning, Gerard, David B. Whyte, Ricardo Martinez, Tony Hunter, and Sucha Sudarsanam. 2002. "The Protein Kinase Complement of the Human Genome." *Science* 298 (5600): 1912–1934. doi:10/fgcbtc.

Marennikova, Svetlana S., Richard C. Condit, and Richard W. Moyer. 2005. *Orthopoxviruses Pathogenic for Humans*. 19–87. New York: Springer. doi:10/dvnpwn.

Marschner, Ian C. 2011. "glm2: Fitting Generalized Linear Models with Convergence Problems." *The R Journal* 3 (2): 12–15.

Masters, John R. 2002. "HeLa Cells 50 Years On: The Good, the Bad and the Ugly." *Nature Reviews Cancer* 2 (4): 315–319. doi:10/d2b2nj.

Matthews, Brian W. 1975. "Comparison of the Predicted and Observed Secondary Structure of T4 phage lysozyme." *Biochimica et Biophysica Acta – Protein Structure* 405 (2): 442–451. doi:10/cs2djx.

McCray, Alexa T., and Nicholas C. Ide. 2000. "Design and Implementation of a National Clinical Trials Registry." *Journal of the American Medical Informatics Association* 7 (3): 313–323. doi:10/cpw3p2.

McCullagh, Peter, and John A. Nelder. 1989. *Generalized Linear Models*. 2nd ed. London: Chapman & Hall.

Mercer, Jason, and Ari Helenius. 2008. "Vaccinia Virus Uses Macropinocytosis and Apoptotic Mimicry to Enter Host Cells." *Science* 320 (5875): 531–535. doi:10/dt66sk.

Misselwitz, Benjamin, Sabrina Dilling, Pascale Vonaesch, Raphael Sacher, Berend Snijder, Markus Schlumberger, Samuel Rout, et al. 2011. "RNAi Screen of *Salmonella* Invasion Shows Role of COPI in Membrane Targeting of Cholesterol and Cdc42." *Molecular Systems Biology* 7:474.1–474.19. doi:10/bkfkkx.

Mohr, Stephanie E., Chris Bakal, and Norbert Perrimon. 2010. "Genomic Screening with RNAi: Results and Challenges." *Annual Review of Biochemistry* 79:37–64. doi:10/cbrgc3.

Mosser, Anne G., Rebecca Brockman-Schneider, Svetlana Amineva, Lacinda Burchell, Julie B. Sedgwick, William W. Busse, and James E. Gern. 2002. "Similar Frequency of Rrhinovirus-Infectible Cells in Upper and Lower Airway Epithelium." *The Journal of Infectious Diseases* 185 (6): 734–43. doi:10/brq9qf.

Nelder, John A, and Robert W.M. Wedderburn. 1972. "Generalized Linear Models." *Journal of the Royal Statistical Society. Series A* 135 (3): 370–384. doi:10/dhq253.

Obbard, Darren J., Karl H.J. Gordon, Amy H. Buck, and Francis M. Jiggins. 2009. "The Evolution of RNAi as a Defence Against Viruses and Transposable Elements." *Philosophical transactions of the Royal Society of London: Series B* 364 (1513): 99–115. doi:10/b93jbd.

Perrimon, Norbert, and Bernard Mathey-Prevot. 2007. "Applications of High-Throughput RNA Interference Screens to Problems in Cell and Developmental Biology." *Genetics* 175 (1): 7–16. doi:10/bpw6hp.

Prussia, Andrew, Pahk Thepchatri, James P. Snyder, and Richard K. Plemper. 2011. "Systematic Approaches Towards the Development of Host-Directed Antiviral Therapeutics." *International Journal of Molecular Sciences* 12 (6): 4027–4052. doi:10/d8f8pw.

R Core Team. 2015. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna. http://www.r-project.org.

Rämö, Pauli, Anna Drewek, Cécile Arrieumerlou, Niko Beerenwinkel, Houchaima Ben-Tekaya, Bettina Cardel, Alain Casanova, et al. 2014. "Simultaneous Analysis of Large-Scale RNAi Screens for Pathogen Entry." *BMC Genomics* 15 (1): 1162.1–1162.18. doi:10/bb4n.

Rämö, Pauli, Raphael Sacher, Berend Snijder, Boris Begemann, and Lucas Pelkmans. 2009. "CellClassifier: Supervised Learning of Cellular Phenotypes." *Bioinformatics* 25 (22): 3028–3030. doi:10/ffgdkk.

Ray, Katrina, Benoit Marteyn, Philippe J. Sansonetti, and Christoph M Tang. 2009. "Life on the Inside: The Intracellular Lifestyle of Cytosolic Bacteria." *Nature Reviews Microbiology* 7 (5): 333–340. doi:10/fp3zbg.

Rieber, Nora, Bettina Knapp, Roland Eils, and Lars Kaderali. 2009. "RNAither, An Automated Pipeline for the Statistical Analysis of High-Throughput RNAi Screens." *Bioinformatics* 25 (5): 678–9. doi:10/cj5pft.

Rocha, Lourena, Luiz Velho, and Paulo Cezar P. Carvalho. 2002. "Image Moments-Based Structuring and Tracking of Objects." In *Proceedings fo the XV Brazilian Symposium on Computer Graphics and Image Processing*, 99–105. Fortaleza: IEEE Computer Society. doi:10/ftvpkk.

Rouilly, Vincent, Eva Pujadas, Bela Hullar, Csaba Balazs, Peter Kunszt, and Michael Podvinec. 2012. "iBRAIN2: Automated Analysis and Data Handling for RNAi Screens." *Studies in Health Technology and Informatics* 175:205–213. doi:10/bb4p.

Saurabh, Satyajit, Ambarish S. Vidyarthi, and Dinesh Prasad. 2014. "RNA Interference: Concept to Reality in Crop Improvement." *Planta* 239 (3): 543–564. doi:10/bb4q.

Schmich, Fabian, Ewa Szczurek, Saskia Kreibich, Sabrina Dilling, Daniel Andritschke, Alain Casanova, Shyan Huey Low, et al. 2015. "gespeR: A Statistical Model for Deconvoluting Off-Target-Confounded RNA Interference Screens." *Genome Biology* 16 (1): 220. doi:10/bb4r.

Schroeder, Gunnar N., and Hubert Hilbi. 2008. "Molecular Pathogenesis of *Shigella* spp.: Controlling Host Cell Signaling, Invasion, and Death by Type III Secretion." *Clinical Microbiology Reviews* 21 (1): 134–156. doi:10/fb28pk.

Schwarz, Hans Rudolf, and Norbert Köckler. 2006. *Numerische Mathematik.* 8th ed. Wiesbaden: Vieweg & Teubner. doi:10/b3qjcq.

Silver, Lynn L. 2011. "Challenges of Antibacterial Discovery." *Clinical Microbiology Reviews* 24 (1): 71–109. doi:10/cmwr5f.

Sittampalam, G Sitta, Nathan P. Coussens, Henrike Nelson, Michelle Arkin, Douglas Auld, Chris Austin, Bruce Bejcek, et al., eds. 2004. *Assay Guidance Manual.* Bethesda: Eli Lilly & Company.

Smith, Alicia. 2012. *Module II: Cellular Infection.* Lecture, Cellular Biochemistry (Part II), Swiss Federal Institute of Technology, Zürich.

Smith, Kevin, Yunpeng Li, Filippo Piccinini, Gabor Csucs, Csaba Balazs, Alessandro Bevilacqua, and Peter Horvath. 2015. "CIDRE: An Illumination-Correction Method for Optical Microscopy." *Nature Methods* 12 (5): 404–406. doi:10/bb4s.

Snijder, Berend, Raphael Sacher, Pauli Rämö, Prisca Liberali, Karin Mench, Nina Wolfrum, Laura Burleigh, et al. 2012. "Single-Cell Analysis of Population Context Advances RNAi Screening at Multiple Levels." *Molecular Systems Biology* 8 (1): 579.1–579.18. doi:10/f2phsb.

Stevens, Joanne M., Edouard E. Galyov, and Mark P. Stevens. 2006. "Actin-Dependent Movement of Bacterial Pathogens." *Nature Reviews Microbiology* 4 (2): 91–101. doi:10/cc3gtf.

Suomalainen, Maarit, Stefania Luisoni, Karin Boucke, Sarah Bianchi, Daniel A. Engel, and Urs F Greber. 2013. "A Direct and Versatile Assay Measuring Membrane Penetration of Adenovirus in Single Cells." *Journal of Virology* 87 (22): 12367–12379. doi:10/bb4t.

Taubenberger, Jeffery, and David M. Morens. 2006. "1918 Influenza: The Mother of All Pandemics." *Emerging Infectious Diseases* Volume 12 (1): 15–22. doi:10/bb4v.

Treadgold, Warren. 1997. *A History of the Byzantine State and Society*. 1044. Stanford: Stanford University Press, June.

Truttmann, Matthias C., Benjamin Misselwitz, Sonja Huser, Wolf-Dietrich Hardt, David R. Critchley, and Christoph Dehio. 2011. "*Bartonella henselae* Engages Inside-Out and Outside-In Signaling by Integrin β_1 and Talin1 During Invasome-Mediated Bacterial Uptake." *Journal of Cell Science* 124 (21): 3591–3602. doi:10/cs6263.

Veiga, Esteban, and Pascale Cossart. 2005. "*Listeria* Hijacks the Clathrin-Dependent Endocytic Machinery to Invade Mammalian Cells." *Nature Cell Biology* 7 (9): 894–900. doi:10/ccszb6.

Venables, William N., and Brian D. Ripley. 2002. *Modern Applied Statistics with S*. 4th ed. New York: Springer. doi:10/bb4w.

Wang, Yejun, He Huang, Ming'an Sun, Qing Zhang, and Dianjing Guo. 2012. "T3DB: An Integrated Database for Bacterial Type III Secretion System." *BMC Bioinformatics* 13 (1): 66. doi:10/bb4x.

Weston, Steve, and Rich Calaway. 2014. *foreach: Foreach Looping Construct for R*. Version Version 1.4.2. Revolution Analytics, Redmond. https://cran.r-project.org/web/packages/foreach.

Weston, Steve, Rich Calaway, and Dan Tenenbaum. 2014. *doParallel: Foreach Parallel Adaptor for the Parallel Package.* Version Version 1.0.8. Revolution Analytics, Redmond. https://cran.r-project.org/web/packages/doParallel.

Whitehead, Kathryn A., Robert Langer, and Daniel G. Anderson. 2009. "Knocking Down Barriers: Advances in siRNA Delivery." *Nature Reviews Drug Discovery* 8 (2): 129–138. doi:10/cxh42p.

Wickham, Hadley. 2007. "Reshaping Data with the reshape Package." *Journal of Statistical Software* 21 (12): 1–20. doi:10/bb42.

Wickham, Hadley. 2014. *Advanced R.* Boca Raton, FL: Taylor & Francis. doi:10/bb45.

Wickham, Hadley. 2015. *R Packages.* North Sebastopol, CA: O'Reilly Media.

Wickham, Hadley, Peter Danenberg, and Manuel Eugster. 2015. *roxygen2: In-Source Documentation for R.* Version Version 4.1.1. RStudio, Boston. https://cran.r-project.org/web/packages/roxygen2.

Wilson, Ross C., and Jennifer A. Doudna. 2013. "Molecular Mechanisms of RNA Interference." *Annual Review of Biophysics* 42 (1): 217–239. doi:10/bb46.

World Health Organization. 2003. *Fact Sheet No. 211: Influenza.* Accessed June 17, 2015. http://www.who.int/mediacentre/factsheets/2003/fs211/en/.

World Health Organization. 2012. *Mortality and Global Health Estimates: Causes of Death.* Accessed June 14, 2015. http://apps.who.int/gho/data/node.main.GHEESTMORT.

World Health Organization. 2014. *Antimicrobial Resistance: Global Report on Surveillance 2014.* Technical report. Geneva: WHO Press. http://www.who.int/drugresistance/documents/surveillancereport.

Yakimovich, Artur, Heidi Gumpert, Christoph J. Burckhardt, Verena A. Lütschg, Andreas Jurgeit, Ivo F. Sbalzarini, and Urs F. Greber. 2012. "Cell-Free Transmission of Human Adenovirus by Passive Mass Transfer in Cell Culture Simulated in a Computer Model." *Journal of Virology* 86 (18): 10123–10137. doi:10/bb5g.

Zhou, Yan-Jun, Jian-Ping Zhu, Tao Zhou, Qun Cheng, Ling-Xue Yu, Ya-Xin Wang, Shen Yang, et al. 2014. "Identification of Differentially Expressed Proteins in Porcine Alveolar Macrophages Infected with Virulent/Attenuated Strains of Porcine Reproductive and Respiratory Syndrome Virus." *PLOS One* 9 (1): e85767. doi:10/bb5h.

Zietz, Björn P., and Hartmut Dunkelberg. 2004. "The History of the Plague and the Research on the Causative Agent *Yersinia pestis*." *International Journal of Hygiene and Environmental Health* 207 (2): 165–178. doi:10/cfk3d7.